Learning with Unlabeled Data

XU, Zenglin

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

January 2009

Thesis/Assessment Committee

Professor Jimmy LEE (Chair)
Professor Irwin King (Thesis Supervisor)
Professor Michael R. Lyu (Thesis Supervisor)
Professor Jun Wang (Committee Member)
Professor James Kwok (External Examiner)

Abstract of thesis entitled:
Learning with Unlabeled Data
Submitted by XU, Zenglin
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in January 2009

We consider the problem of learning from both labeled and unlabeled data through the analysis on the quality of the unlabeled data. Usually, learning from both labeled and unlabeled data is regarded as semi-supervised learning, where the unlabeled data and the labeled data are assumed to be generated from the same distribution. When this assumption is not satisfied, new learning paradigms are needed in order to effectively explore the information underneath the unlabeled data. This thesis consists of two parts: the first part analyzes the fundamental assumptions of semi-supervised learning and proposes a few efficient semi-supervised learning models; the second part discusses three learning frameworks in order to deal with the case that unlabeled data do not satisfy the conditions of semi-supervised learning.

In the first part, we deal with the unlabeled data that are in good quality and follow the conditions of semi-supervised learning. Firstly, we present a novel method for Transductive Support Vector Machine (TSVM) by relaxing the unknown labels to the continuous variables and reducing the non-convex optimization problem to a convex semi-definite programming problem. In contrast to the previous relaxation method which involves $\mathcal{O}(n^2)$ free parameters in the semi-definite matrix, our method takes

advantage of reducing the number of free parameters to $\mathcal{O}(n)$, so that we can solve the optimization problem more efficiently. In addition, the proposed approach provides a tighter convex relaxation for the optimization problem in TSVM. Empirical studies on benchmark data sets demonstrate that the proposed method is more efficient than the previous semi-definite relaxation method and achieves promising classification results comparing with the state-of-the-art methods. Our second contribution is an extended level method proposed to efficiently solve the multiple kernel learning (MKL) problems. In particular, the level method overcomes the drawbacks of both the Semi-Infinite Linear Programming (SILP) method and the Subgradient Descent (SD) method for multiple kernel learning. Our experimental results show that the level method is able to greatly reduce the computational time of MKL over both the SD method and the SILP method. Thirdly, we discuss the connection between two fundamental assumptions in semi-supervised learning. More specifically, we show that the loss on the unlabeled data used by TSVM can be essentially viewed as an additional regularizer for the decision boundary. We further show that this additional regularizer induced by the TSVM is closely related to the regularizer introduced by the manifold regularization. Both of them can be viewed as a unified regularization framework for semi-supervised learning.

In the second part, we discuss how to employ the unlabeled data for building reliable classification systems in three scenarios: (1) only poorly-related unlabeled data are available, (2) good quality unlabeled data are mixed with irrelevant data and there are no prior knowledge on their composition, and (3) no unlabeled data are available but can be achieved from the Internet for text categorization. We build several frameworks to deal with the above cases. Firstly, we present a study on how to deal with the weakly-related unlabeled data, called the

Supervised Self-taught Learning framework, which can transfer knowledge from the unlabeled data actively. The proposed model is able to select those discriminative features or representations, which are more appropriate for classification. Secondly, we also propose a novel framework that can learn from a mixture of unlabeled data, where good quality unlabeled data are mixed with unlabeled irrelevant samples. Moreover, we do not need the prior knowledge on which data samples are relevant or irrelevant. Consequently it is significantly different from the recent framework of semi-supervised learning with universum and the framework of Universum Support Vector Machine. As an important contribution, we have successfully formulated this new learning approach as a Semi-definite Programming problem, which can be solved in polynomial time. A series of experiments demonstrate that this novel framework has advantages over the semi-supervised learning on both synthetic and real data in many facets. Finally, for third scenario, we present a general framework for semi-supervised text categorization that collects the unlabeled documents via Web search engines and utilizes them to improve the accuracy of supervised text categorization. Extensive experiments have demonstrated that the proposed semi-supervised text categorization framework can significantly improve the classification accuracy. Specifically, the classification error is reduced by 30% averaged on the nine data sets when using Google as the search engine.

在未標記的數據中的機器學習

徐增林

本論文探討如何從廣義的未標記數據中進行學習的研究。通常針對未標記數據的學習被稱為半監督的學習(semi-supervised learning)。半監督的學習假設未標記的數據和標記的數據產生於同一個分佈。因此，當這個假設不成立時，就需要新的學習方式來有效發覺隱含在未標記 數據中的信息。本文從相應的兩個角度回答了上述兩種不同情形下的學習中的問題。本文由兩部分組成：第一部分分析了半監督學習的基本假設，並提出了高效的半監督學習算法，第二部分針對未標記數據和標記數據由不同分佈產生的具體情形提出了三個不同的學習框架。

第一部分，我們考慮有大量未標記的數據具有優良品質的情形。首先，我們提出了一種新的直推式支持向量機(Transductive Support Vector Machine, TSVM)學習方法。在這個方法中，我們將離散變量放鬆為連續變量，從而把非凸優化問題簡化為凸的半定規劃(Semi-definite Programming, SDP)問題。跟之前的具有 O(n^2)個自由變量的方法相比較，本文提出的算法只有 O(n)個變量，因而大大提高了相應優化問題的效率。另外，本文提出 的方法對直推式支持向量機提供了一個更緊緻的解。實驗結果表明本文提出的算法比之前的放鬆式且直推式支持向量機更高效而且取得了非常有前景的結果。本文的另 一個貢獻是一種用於解決多重核學習(multiple kernel learning, MKL)的水平(level)方法。該方法的克服了傳統半無限線性規劃(semi-infinite linear programming, SILP)和次梯度下降(Subgradient Descent, SD)方法用於多重核學習時的缺陷。實驗結果表明我們提出的方法能夠極大地減少多重核學習的計算代價。再次，我們討論了半監督學習中的兩種基本假設之間的 聯繫。具體來講，直推式支持向量機中未標記數據上的損失函數本質上可以看作是對決策邊界的正則化。我們進一步表明直推式支持向量機中的正則化項跟流形正則 化(manifold regularization)密切相關。二者可以看作是半監督學習中的一個統一的正則化框架。

第二部分，我們考慮在三種不同的情形下如何利用未標記數據建立可靠的分類系統：(1)只有弱相關的數據可用，(2)品質良好的數據同不相關數據混 和在一起而且沒有先驗知識可用，(3)沒有未標記數據可用但是可以設計查詢算法主動從 Internet 上獲取。對上述不同情形，我們分別建立相應的學習框 架。首先，我們提出了一個關於如何從弱相關數據中學習的框架。我們稱之為有監督的自學習框架(supervised self-taught learning),它可以主動從未標記數據中轉移知識。我們提出的模型能夠提取那些具有判別性的更有利於分類的特徵。其次，我們提出了一個新的可以用於 從優質數據和無關數據的混和體中進行學習的框架。而且我們不需要任何關於哪些數據是相關哪些是無關的先驗知識。因此，本框架顯著區別於之前的使用 Universum 的半監督學習和 Universum 支持向量機。作為一個重要的貢獻，我們把該學習模型表示為半定規劃問題，因而可以在多項式時間內求 解。在人工和實際數據上的一系列實驗表明了這個新框架的多方面的優越性。最後，對於上述第三種情形，我們提出了一個針對半監督的文本分類的通用框架，它可 以通過搜索引擎來從 Internet 上獲取未標記文本，從而提高有監督的文本分類的準確率。詳細的實驗表明，本文提出的半監督文本分類框架可以極大地提高 文本分類的準確率。具體來講，當使用 Google 作為搜索引擎時，平均能夠將準確率提高 30%。

Acknowledgement

I would like to express my gratitude to all those who have helped in accomplishing this thesis. First and foremost, I would like to thank my supervisors, Professor Irwin King and Professor Michael R. Lyu. I gain too much from their guidance not only on doing research but also on the presentation, teaching, and English writing skills. I would like to express my sincere gratitude and appreciation to their supervision, encouragement, and support at all levels. I also extend warm thanks to Prof. Rong Jin, from whom, I have received many valuable suggestions and comments in conducting research in machine learning. We have collaborations on several pieces of great work. I will always be grateful for the outstanding research environment fostered by our department.

I am also grateful to my colleagues and my friends. I thank Dr. Kaizhu Huang and Mr. Jianke Zhu for their effort and constructive discussions in conducting the research work in this thesis. I also want to thank my colleges in the machine learning group and the social network group, Haixuan Yang, Haiqin Yang, Hao Ma, Hongbo Deng, Zhenjiang Lin, Tomas Chan, Patrick Lau, and Wei Wei.

Last but not least, I want to thank my wife and my parents. Without their deep love and constant support, this thesis would never have been completed.

Contents

Abstract **i**

Acknowledgement **v**

1 Introduction **1**

1.1 Efficient and Effective Models Employing Unlabeled Data for Semi-supervised Learning 4

1.2 Efficient Multiple Kernel Learning 4

1.3 Unified View on Assumptions of Semi-supervised Learning 5

1.4 Exploring Weakly-related Unlabeled Data 6

1.5 Learning from a Mixture of Unlabeled Data . . . 7

1.6 Actively Searching Unlabeled Data 8

1.7 Contributions 9

1.8 Scope 13

1.9 Thesis Organization 14

2 Background Review: Learning with Unlabeled Data **17**

2.1 Supervised Learning 20

2.2 Unsupervised Learning 23

2.3 Semi-supervised Learning 24

2.3.1 Transductive Support Vector Machine . . 27

2.3.2 Graph-based Semi-supervised Learning Models 28

2.3.3 Other Semi-supervised Models 28

2.4 Learning from Variance-shifted Unlabeled Data . 29
2.5 Learning from Weakly-related Data 30
2.6 Learning with Universum 31
2.7 Kernel Learning 32
2.7.1 Supervised Kernel Learning 32
2.7.2 Semi-supervised Kernel Learning 33
2.8 Semi-supervised Text Categorization 34
2.8.1 Related Work on Text Categorization . . . 34
2.8.2 Related Work on Query Generation 36

3 Efficient Convex Relaxation for TSVM 38
3.1 Convex Relaxation of TSVM 40
3.2 Efficient Convex Relaxed Transductive Support Vector Machine 42
3.2.1 Hard Margin TSVM 42
3.2.2 Soft Margin TSVM 47
3.3 Experiments 48
3.3.1 Data Sets Description 48
3.3.2 Experimental Protocol 49
3.3.3 Experimental Results 51
3.4 Summary and Future Work 52

4 Level Method for Efficient Multiple Kernel Learning 53
4.1 Multiple Kernel Learning 55
4.2 Extended Level Method for MKL 57
4.2.1 Introduction to the Level Method 57
4.2.2 Extension of the Level Method to MKL . . 58
4.2.3 Semi-supervised Multiple Kernel Learning 63
4.3 Experiments 64
4.3.1 Experiments on supervised MKL 64
4.3.2 Experiments on Semi-supervised MKL . . 67
4.4 Summary and Future Work 69

5 Unified Framework for Semi-supervised Learning 72
5.1 TSVM: A Regularization View 73
5.1.1 Learning Model when $\rho = 0$ 75
5.1.2 Learning Model when ρ is small 76
5.1.3 Learning Model when ρ is large 77
5.2 Understanding Manifold Regularization 79
5.3 Summary 80

6 Learning from Weakly-related Unlabeled Data 81
6.1 Problem Formalism 85
6.2 Self-taught Learning by Sparse Coding 86
6.2.1 Stage I: Learning Representations 86
6.2.2 Stage II: Feature Construction from Basis 87
6.2.3 Stage III: Learning a Classifier from Features 87
6.3 Supervised Self-taught Learning Framework . . . 88
6.3.1 Two-category Model 89
6.3.2 Optimization Method 90
6.3.3 Multi-category Model 91
6.4 Experiments 93
6.4.1 Toy Example 93
6.4.2 Handwritten numerals → English Characters 95
6.4.3 Web Text Categorization 96
6.5 Discussion 98
6.6 Summary 100

7 Learning from a Mixture of Unlabeled Data 102
7.1 Model 106
7.1.1 Problem Formalism 106
7.1.2 Framework 107
7.2 Experiment 114
7.2.1 Evaluation on Synthetic Data 115

7.2.2 Evaluation on Real Data 118
7.3 Summary 120

8 Semi-supervised Learning by Active Search 122
8.1 Semi-supervised Text Categorization by Active Search 126
8.1.1 Problem Definition 127
8.1.2 Discriminative Query Generation (DQG) . 128
8.1.3 Text Categorization Methods 131
8.2 Experiment 135
8.2.1 Data Set 136
8.2.2 Evaluation (I): Query Generation 137
8.2.3 Evaluation (II): Text Categorization Methods 141
8.2.4 Evaluation (III): Impact of Search Engines 143
8.3 Summary and Future Work 144

9 Conclusion and Future Work 148
9.1 Review of the Journey 149
9.2 Future Work 151

A Proof of Theorem 1, Chapter 4 153

B Publications 156

Bibliography 158

List of Figures

2.1 The illustration of different types of unlabeled data. 19

2.2 A categorization of unlabeled data and their related learning paradigms. 20

2.3 An illustration of Support Vector Machine 23

2.4 The illustration of two semi-supervised learning methods based on low density assumption and manifold assumption, respectively. 26

2.5 The relationship among semi-supervised learning, low density assumption, manifold assumption and their motivated methods. 26

3.1 Computation time of the proposed convex relaxation approach for TSVM (i.e., CTSVM) and the semi-definite relaxation approach for TSVM (i.e., RTSVM) versus the number of unlabeled examples. The Course data set is used, and the number of labeled examples is 20. 50

4.1 Illustration of the Level method. We aim to minimize $f(x)$ over [-4,4]. With the help of the affine lower bound function $g_i(x)$, we are able to gradually approximate the optimal solution. ∇ denotes the lower bound value in one iteration. 59

4.2 Evolution of objective values over time (seconds) for datasets "Iono", "Breast", and "Pima". The objective values are plotted on a logarithm scale (base 10) for better comparison. Only parts of the evolution curves are plotted for SILP due to their long tails. 67

4.3 The evolution curves of the five largest kernel weights for datasets "Iono", "Breast" and "Pima" computed by the three MKL algorithms 68

6.1 High-level visual features extracted by STL and SSTL when classifying digits "1" and "7" with capital letters "I", "M", "N" as unlabeled samples. (a) and (b) presents the patterns learned by STL and SSTL respectively. In (a), STL fail to extract the discriminative features, i.e., horizontal stroke patterns. In (b), SSTL manages to learn many horizontal stroke patterns. 83

6.2 Examples of labeled data and unlabeled data used in the toy example. The top sub-figure contains 4 numerals, labeled as "1" for the first two characters, and "7" for the last two numerals. The bottom sub-figure provides 20 examples randomly extracted from 200 28×28 unlabeled English characters representing "I", "M", and "N". The basis number is set to 50. λ, γ, β, are all set to 10. . . 94

7.1 The "irrelevant" data $\star$'s can increase the performance of the SSL. The filled points ($\bullet$'s and $\star$'s) are unlabeled data, while the $\circ$'s and $\Box$'s are the two classes of labeled training samples. The filled $\star$'s describe the irrelevant unlabeled data. The decision planes of the SL and SSL are given by the SVMs. 104

7.2 The "irrelevant" data $\star$'s can increase the performance when only a limited number of relevant unlabeled data is available. The filled points ($\bullet$'s and $\star$'s) are unlabeled data, while the $\circ$'s and $\Box$'s are the two classes of labeled training samples. The filled $\star$'s describe the irrelevant unlabeled data. The decision plane of the SSL is given by the SVM. 105
7.3 Hinge loss and ε-insensitive loss 109
7.4 Comparison of SSL and the proposed USSL on the synthetic data 115
7.5 Comparison of SSL and the proposed model USSL on synthetic data. (a)-(c) plot the training data for the three data sets respectively. (d)-(f) plot the decision boundary given by SSL as well as the class label of the unlabeled data assigned by SSL. (g)-(i) plot the decision boundary given by USSL as well as the class label of the unlabeled data assigned by USSL. (j)-(l) show the results on test data. The proposed USSL generates more reasonable decision boundaries and outperforms the traditional SSL. 116

8.1 The framework of semi-supervised text categorization by active search. 124
8.2 The classification accuracy of text categorization methods (i.e., Auxi-SVM and Semi-SVM) using different search engines (i.e., Google, Yahoo!, and Alltheweb) on three data sets of Ohmued 143

List of Tables

3.1 Data sets used in the experiments, where d represents the data dimensionality, l means the number of labeled data points, and n denotes the total number of examples. 49

3.2 The classification performance of Transductive SVMs on benchmark data sets. 51

4.1 The performance comparison of three MKL algorithms. Here n and m denote the size of training samples and the number of kernels, respectively. . . 70

4.2 Time-saving ratio of the level method over the SILP and the SD method 71

4.3 The performance comparison of three MKL algorithms for semi-supervised learning. 71

6.1 Experimental results on classifying “1” and “7” with unlabeled characters “I”, “M”, “N” 94

6.2 Experimental results on classification of 26 English characters with unlabeled handwritten numerals data 96

6.3 Descriptions for the web text documents data . . 97

6.4 Comparisons on web text categorization tasks. STL performs worse than SVM usually due to the inappropriate high-level representations learned. SSTL presents the best results consistently by incorporating the knowledge selectively and discriminatively. 98

7.1 Experimental results on USPS data 120
7.2 Experimental results on MNIST data 120

8.1 Data sets for evaluation. 137
8.2 The query terms selected by different query generation methods for the data set *bacterial vs. virus.* The query size is set to be three for all methods. . 146
8.3 The query terms selected by different query generation methods for the data set *money vs. trade.* The query size is set to be three for all methods. . 146
8.4 The retrieval precision of different query generation methods with varied query sizes. 147
8.5 The classification accuracy (%) semi-supervised text categorization methods 147

Chapter 1

Introduction

Machine learning is a subfield of artificial intelligence that is concerned with the design and development of algorithms and techniques that allow computers to make inductions or deduction [102]. In general, machine learning studies a variety of different types of problems. In terms of the different settings and ways of dealing with data, machine learning algorithms can typically be categorized as unsupervised learning, supervised learning, semi-supervised learning, and reinforcement learning, and others. We give a simple description of these learning algorithms in the following:

- Supervised learning that generates a function that maps inputs to desired outputs. In supervised learning, each instance of the training data consists of a data vector and it corresponding output. In terms of the output, there are two supervised supervised learning tasks: classification where the output is discrete and regression where the output is continuous.

- Unsupervised learning employs only the unlabeled examples where no category information is available. One typical unsupervised learning task is data clustering.

- Semi-supervised learning combines both labeled and unlabeled examples to generate an appropriate function or classifier.

- Reinforcement learning algorithms learn a policy of how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback that guides the learning algorithm.

One of the major problems in machine learning is how to get a large amount of labeled examples. However, data labeling is usually expensive due to the fact that labeling requires a lot of human efforts. While unlabeled data could be relatively easy to obtain. For example, it is easy to download a batch of web pages from the internet, but it requires experts to label the pages. Therefore, semi-supervised learning, which employs both the labeled data and unlabeled data, has attracted a lot of research focus in recently years. One of the important issues in semi-supervised learning is how to efficiently and effectively explore the information underneath the unlabeled data.

The objective of this thesis is to establish a framework that effectively and efficiently employs the information underlying unlabeled data, given a small amount of labeled data and a large amount of unlabeled data, through the analysis on the unlabeled data. Since labeled data are usually expensive to obtain and unlabeled data are relatively easy to obtain, unlabeled data has recently attracted the research focus in machine learning [28, 163, 140, 112]. One typical learning paradigm of employing the unlabeled data is semi-supervised learning [28, 163], which assumes that the unlabeled data and the labeled data are generated by the same distribution. Semi-supervised learning have achieved successes in many applications, such as text categorization, image classification, and spam filtering. When there are no high-quality unlabeled data, semi-supervised learning cannot be employed to completely explore the information underneath the unlabeled data. It has been shown in [112] that weakly-related unlabeled data which may have different class labels as the training data can also be utilized to improve the prediction accuracy in the small-training-sample scenarios. Researchers also demonstrate that irrelevant data or background data can also be helpful for constructing good prediction functions [140].

This thesis discusses how to employ the unlabeled data for building reliable classification systems in different scenarios: (1) good quality unlabeled data are available, (2) only poorly-related unlabeled data are available, (3) good quality unlabeled data are mixed with irrelevant data but with no prior knowledge on their composition, and (4) unlabeled data are available in the Internet for some specific application such as text categorization. We build several frameworks to deal with the above cases. More specifically, the resulting principled framework includes efficient models unifying the underlying assumptions in semi-supervised learning, models effectively exploring the information behind weakly-related unlabeled data, models dealing with the mixture of different kinds of unlabeled data, and models actively searching unlabeled data from the Internet with applications to text categorization.

In this chapter, we address the motivations of the framework of learning from unlabeled data. We present the objectives of this thesis and outline the contributions. Finally, we provide an overview of the rest of this thesis.

1.1 Efficient and Effective Models Employing Unlabeled Data for Semi-supervised Learning

Semi-supervised learning has attracted an increasing amount of research interest recently [29, 163]. An important semi-supervised learning paradigm is the Transductive Support Vector Machine (TSVM), which maximizes the margin in the presence of unlabeled data and keeps the boundary traversing through low density regions, while respecting labels in the input space. We consider the problem of Support Vector Machine transduction, which involves a combinatorial problem with exponential computational complexity in the number of unlabeled examples. Although several studies are devoted to Transductive SVM, they suffer either from the high computation complexity or from the solutions of local optimum. To address this problem, we propose solving Transductive SVM via convex relaxation, which converts the NP-hard problem to a semi-definite programming. Compared with the other SDP relaxation for Transductive SVM, the proposed algorithm is computationally more efficient with the number of free parameters reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, where n is the number of examples. An empirical study with several benchmark data sets shows the promising performance of the proposed algorithm in comparison with other state-of-the-art implementations of Transductive SVM.

1.2 Efficient Multiple Kernel Learning

Kernel learning [86, 109, 100] has received much attention in the machine learning communities in recent years. This is due to the importance of kernel methods in that kernel functions define a generalized similarity measure among data. A generic approach to learning a kernel function is known as multiple kernel learning (MKL) [86]: given a list of base kernel functions/matrices, MKL searches for the linear combination of base kernel functions which maximizes a generalized performance measure.

We consider the problem of multiple kernel learning, which can be formulated as a convex-concave problem. In the past, two efficient methods, i.e., Semi-Infinite Linear Programming (SILP) and Subgradient Descent (SD), have been proposed for large-scale multiple kernel learning. Despite their success, both methods have their own shortcomings: (a) the SD method utilizes the gradient of only the current solution, and (b) the SILP method does not regularize the approximate solution obtained from the cutting plane model. In this work, we extend the level method, which was originally designed for optimizing non-smooth objective functions, to convex-concave optimization, and apply it to multiple kernel learning. The extended level method overcomes the drawbacks of SILP and SD by exploiting all the gradients computed in past iterations and by regularizing the solution via a projection to a level set. An empirical evaluation with eight UCI datasets shows that the extended level method can significantly improve efficiency by saving on average 91.9% of computational time over the SILP method and 70.3% over the SD method.

1.3 Unified View on Assumptions of Semi-supervised Learning

Generally, semi-supervised learning methods are derived from two fundamental geometric assumptions: the low density assumption (or cluster assumption) and the manifold assumption.

In the low density assumption, decision boundaries should not cross high density regions, but instead lie in low density regions. Semi-supervised learning methods based on this assumption usually minimize the losses on both the labeled and unlabeled data. Typical methods include label-switching-retraining [75], replacing the loss functions [37, 34], and convex relaxation methods [145, 148].

In the manifold assumption, data are assumed to form a low-dimensional manifold in some input space. Many semi-supervised methods implement such an assumption by using the graph Laplacian of a graph-based representation to characterize the manifold structure. Typical methods include semi-supervised spectral kernel learning [165], semi-supervised learning using gaussian fields [164], the point-cloud kernel [125], learning with local and global consistency [160], manifold regularization [10], etc.

Although these types of two approaches are based on different motivations, they essentially share similar spirit, namely the decision boundary should be decided by not only the labeled examples, but also the structure of the unlabeled examples. In the framework of transductive SVM, the regularization of decision boundary by the unlabeled data is achieved by the minimization of the loss function for the unlabeled data. In contrast, the manifold regularization approach regulates the choice of decision boundary by an additional term of regularizer that is constructed by the Laplacian of the unlabeled data. We show that the unlabeled data used by TSVM can essentially be viewed as an additional regularizer for the decision boundary. We further show that this additional regularizer induced by the TSVM is closely related to the regularizer introduced by the manifold regularization.

1.4 Exploring Weakly-related Unlabeled Data

We consider the task of learning from weakly-related unlabeled data which may not share the same category labels as the labeled data. This task is also called self-taught learning (STL)

[112]. In contrast to Semi-supervised learning that requires unlabeled data to share the same category labels as the labeled data, STL can transfer knowledge from very different unlabeled data that share only similar structural information with the labeled data. STL generally exploits a three-step strategy: (1) learning high-level representations from unlabeled data only (2) re-constructing the labeled data by such representations, and (3) building a classifier over the re-constructed labeled data. However, the learned knowledge, i.e., the high-level representations, exclusively determined by the unlabeled data, may be inappropriate or even misleading for the latter classifier learning step. In this thesis, we propose a novel Supervised Self-taught Learning (SSTL) framework that successfully integrates the optimization of the three steps of STL into just one step. By integrating the process of classifier optimization with that of choosing the high-level representations, the proposed model focuses on selecting those discriminant representations, which are more appropriate for classification. One important feature of our novel framework is that the final optimization can be iteratively solved with the convergence guaranteed. We evaluate our novel framework on various data sets. The experimental results show that the proposed SSTL outperforms STL and the traditional supervised learning methods.

1.5 Learning from a Mixture of Unlabeled Data

We consider the problem of Semi-supervised Learning (SSL) from a mixture of unlabeled data, which may contain irrelevant samples. Within the binary setting, our model manages to better utilize the information from unlabeled data by formulating them as a three-class $(-1, +1, 0)$ mixture, where class 0 represents the irrelevant data. This distinguishes our work from the traditional SSL problem where unlabeled data are assumed to contain relevant samples only, i.e., either +1 or −1, which are forced to be the same as the given labeled samples. This work is also different from another learning paradigm, i.e., learning with universum (universum means "irrelevant" data), in that the universum need not to be specified beforehand in our work. Indeed, one of the significant contributions of our proposed framework is that such irrelevant samples can be automatically detected from the available unlabeled data. This hence presents a general SSL framework that does not force "clean" unlabeled data. More importantly, we formulate this general learning framework as a Semi-definite Programming problem, making it solvable in polynomial time. A series of experiments demonstrate that the proposed framework can outperform the traditional SSL on both synthetic and real data.

1.6 Actively Searching Unlabeled Data

The goal of automated text categorization is to automatically classify documents into predefined categories. This task is usually studied as a supervised problem where a statistical model is learned from a pool of labeled documents. However, given a small number of labeled documents, it is very challenging, if not impossible, to build a reliable classifier that is able to achieve high classification accuracy; this small-size sample problem is commonly seen in Web applications due to the high costs in manually labeling documents. To address this problem, a novel Web-assisted text categorization framework is proposed. Important keywords are first automatically identified from the available labeled documents to form the queries. Search engines such as Google or Yahoo! are then utilized to retrieve from the Web a multitude of relevant documents. A semi-supervised framework is finally engaged to exploit both labeled samples and returned unlabeled documents for building a more accurate classifier. Unlike most semi-supervised learning algorithms that assume the unlabeled documents are available, the proposed framework actively seeks the relevant documents by "asking" the expert (i.e., search engines) with carefully designed queries. To our best knowledge, this work is the first study of this kind. As a key contribution, we elegantly formulate the query generation task as a learning problem that aims to extract the most discriminative keywords for search from the labeled documents at hand. An extensive experimental study shows the encouraging results of the proposed text categorization framework: using Google as the Web search engine, the proposed framework is able to reduce the classification error by 30% when compared with the state-of-the-art supervised text categorization method.

1.7 Contributions

In this thesis, we aim to propose a general framework to efficiently and effectively exploring the information behind the unlabeled data based on the property or quality of unlabeled data. Within this framework, the thesis consists of two parts: the first part deals with good quality unlabeled data as often used in semi-supervised learning literatures, and the second part deals with general unlabeled data which may be not drawn from the same distribution. In the first part, we firstly propose an efficient convex relaxation model of Transductive SVM. Furthermore, we propose an efficient multiple kernel learning approach and naturally extend it to semi-supervised learning. We then discuss the relationship between the low-density assumption and the manifold assumption for semi-supervised learning. In the second part, we relax the constraint of the quality of unlabeled data. We first consider a setting that the unlabeled data are only structurally-related and may not share the same label with the training data. We then consider another setting that irrelevant data are mixed with good quality data. Finally, we explore the possibility to actively search for unlabeled data from the Internet for semi-supervised learning with its application to text categorization. The main contributions of this thesis are further described as follows in detail.

- **Proposing an efficient convex relaxation model for Transductive SVM (published in NIPS2007)**

  - ◇ Unlike the semi-definite relaxation [145] that approximates TSVM by dropping the rank constraint, the proposed approach approximates TSVM by its dual problem. As the basic result of convex analysis, the conjugate of conjugate of any function $f(\mathbf{x})$ is the convex envelope of $f(\mathbf{x})$, and therefore provides a tighter convex relaxation for $f(\mathbf{x})$ [57]. Hence, the proposed approach provides a better convex relaxation than that in [145] for the optimization problem in TSVM.
  - ◇ Compared with the semi-definite relaxation TSVM, the proposed algorithm involves fewer free parameters and therefore significantly improves the efficiency by reducing the worst-case computational complexity from $\mathcal{O}(n^{6.5})$ to $\mathcal{O}(n^{4.5})$.

- **Proposing an efficient method for multiple kernel learning (published in NIPS2008)**

  - ◇ We discuss the level method, which was originally designed for optimizing non-smooth objective functions, to convex-concave optimization. We apply it to multiple kernel learning. The extended level method overcomes the drawbacks of SILP and SD by exploiting all the gradients computed in past iterations and by regularizing the solution via a projection to a level set.

- ◇ We propose an efficient semi-supervised multiple kernel learning approach by deforming each base kernel matrix with the graph laplacian. We therefore solve the problem of finding the best parameter for selecting the kernel for deforming in the semi-supervised setting.

- **Proposing a framework that unifies two important assumptions in semi-supervised learning**
  - ◇ We discuss the relationship between low density assumption and manifold assumption by considering two implementations of them, i.e., transductive SVM (TSVM), which is based on low density assumption, and the approach of manifold regularization, which is based on manifold assumption. In the framework of TSVM, the regularization of decision boundary is achieved by minimizing the loss on unlabeled data. On the other hand, the manifold regularization approach regulates the choice of decision boundary by additional regularizer that is constructed by graph Laplacian regularization.
  - ◇ We theoretically prove that the loss on unlabeled data in TSVM can be regarded as a special graph Laplacians.
  - ◇ We formulate the manifold regularizer as a regularization term of TSVM.

- **Proposing a novel Supervised Self-taught Learning (SSTL) model**
  - ◇ The proposed model manages to find the most appropriate high-level features or representations from the unlabeled data under the supervision of the labeled training data. We attempt to learn from unlabeled

data with the “target” in mind rather than to achieve it in a hit-or-miss way.

- ◇ The three stages (the basis learning, the coefficient optimization, and the classifier learning) are integrated into a single optimization problem. The representations, the coefficients, and the classifier are optimized simultaneously. By interacting the classifier optimization with choosing the high-level representations, the proposed model is able to select those discriminant features or representations, which are most appropriate for classification. Hence it will greatly benefit the classification performance.

- **Proposing a general framework for learning from a mixture of unlabeled data (published in ICDM2008)**
  - ◇ We propose a framework of Semi-supervised Learning (SSL) from a mixture of unlabeled data, which may contain irrelevant samples. This framework avoids the requirement of the prior knowledge on the composition of the unlabeled data. This distinguishes our work from the traditional SSL problem where unlabeled data are assumed to contain relevant samples only. This work is also different from another family of popular models, universum learning (universum means “irrelevant” data), in the sense that the universum need not to be specified beforehand.
  - ◇ We formulate this general learning framework as a Semidefinite Programming problem, making it solvable in polynomial time.

- **Proposing a general framework for semi-supervised text categorization via active search (published in CIKM2008)**

- ◇ We discuss in detail a general framework for active semi-supervised text categorization that collects the unlabeled documents via Web search and utilizes them to improve the accuracy of supervised text categorization.
- ◇ We present a novel learning approach, named Discriminative Query Generation (DQG) method, for query generation that improves the chance of finding the documents relevant to the target topics via Web retrieval. Both theoretical justifications and empirical evaluations demonstrate that the DQG approach significantly outperforms other intuitive methods such as Term Frequency (TF) [53], Term Frequency/Inverse Document Frequency (TF/IDF) [20], and Odds-ratio [53].
- ◇ We engage the semi-supervised learning method to perform text categorization that can effectively exploit both the labeled documents and the unlabeled Web documents which are retrieved by Web search engines. Extensive results show that semi-supervised learning framework is consistently superior to the purely supervised method and the supervised method with auxiliary text.

1.8 Scope

This thesis states and refers to the learning first as statistical learning, which appears to be the current main trend of learning approaches. We then further restrict the learning in the framework of semi-supervised learning, one of the main problems in machine learning. This thesis is also related to the machine learning techniques dealing with the unlabeled data. The corresponding discussion on different models including the conducted analysis of the computational and statistical aspects of machine learning are all subject to the classification tasks with the availability of unlabeled data.

1.9 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2
  We first categorize the unlabeled data into several types. We will review different learning paradigms based on the data types, including semi-supervised learning, transfer learning, etc, in this chapter. We will first review semi-supervised learning from their motivations, i.e., low density assumption or manifold assumption. We will then relax the requirement of unlabeled data from the same distribution to a variational distribution, and then to a totally irrelevant distribution. This leads to other techniques, such as transfer learning and self-taught learning, of using unlabeled data.

- Chapter 3
  We will develop a novel efficient relaxation model for Transductive Support Vector Machine (TSVM). We will demonstrate how this new model provides a tighter and more efficient approximation than previous SDP relaxation. We will then present a series of experiments to demonstrate the advantages of this model.

- Chapter 4
  We develop an extended level method which is one of the recent advances in optimization for constructing an efficient multiple kernel learning approach. We will show its advantage theoretically and empirically.

- Chapter 5
  We will discuss a unified view on the assumptions under-

lying semi-supervised learning. We will theoretically show how these two assumptions can be connected together by using TSVM as an example.

- Chapter 6
  We will develop a novel model which manages to find the most appropriate high-level features or representations from the poorly-related unlabeled data under the *supervision* of the good-quality labeled training data. This model is also called Supervised Self-taught Learning (SSTL). We will show that the resulting model can be formulated in one single optimization problem with guaranteed convergence. Both illustrations on toy data sets and evaluations on real world data sets will be provided in this chapter.

- Chapter 7
  We will discuss the problem of learning from the mixture of unlabeled data, which may contain irrelevant samples. We will propose a general framework dealing with the mixture of relevant labeled data and irrelevant unlabeled data. This hence presents a general semi-supervised learning framework that does not force “clean” unlabeled data. We further conduct a series of experiments to demonstrate that the proposed framework can outperform the traditional semi-supervised learning on both synthetic and real data.

- Chapter 8
  We discuss a general framework for self-taught text categorization, which collects the unlabeled documents via Web search engines and utilizes them to improve the accuracy of supervised text categorization. We will first generate high quality query terms for a search engine based on a small number of given training documents. The downloaded documents generated by the search engine will then be used as unlabeled data for semi-supervised learning methods or

auxiliary methods. Empirical evaluations on several benchmark data sets will be presented to demonstrate the merits of our proposed semi-supervised text categorization framework by active search.

- Chapter 9
  We will then summarize this thesis and conduct discussions on future work.

In order to make each of these chapters self-contained, some critical contents, e.g., model definitions or motivations having appeared in previous chapters, may be briefly reiterated in some chapters.

□ **End of chapter.**

Chapter 2

Background Review: Learning with Unlabeled Data

In this chapter, we conduct a more detailed and more formal review on the techniques about how to employ unlabeled data, following a brief review of traditional supervised statistical learning methods. In order to clearly show the relationship among existing techniques for learning with unlabeled data, we classify the types of unlabeled data into five categories. To better understand the properties of different types of unlabeled data, we introduce the classification task of elephant images and rhino images. We show the labeled images of this task in Figure 2.1 (a).

We summarize the types of unlabeled data in the following:

- Type I: the unlabeled data and the labeled data are drawn from the same distribution. The related learning frame work is semi-supervised learning [163, 28]. For the classification task of elephants and rhinos, the unlabeled data can be the images in Figure 2.1 (b).

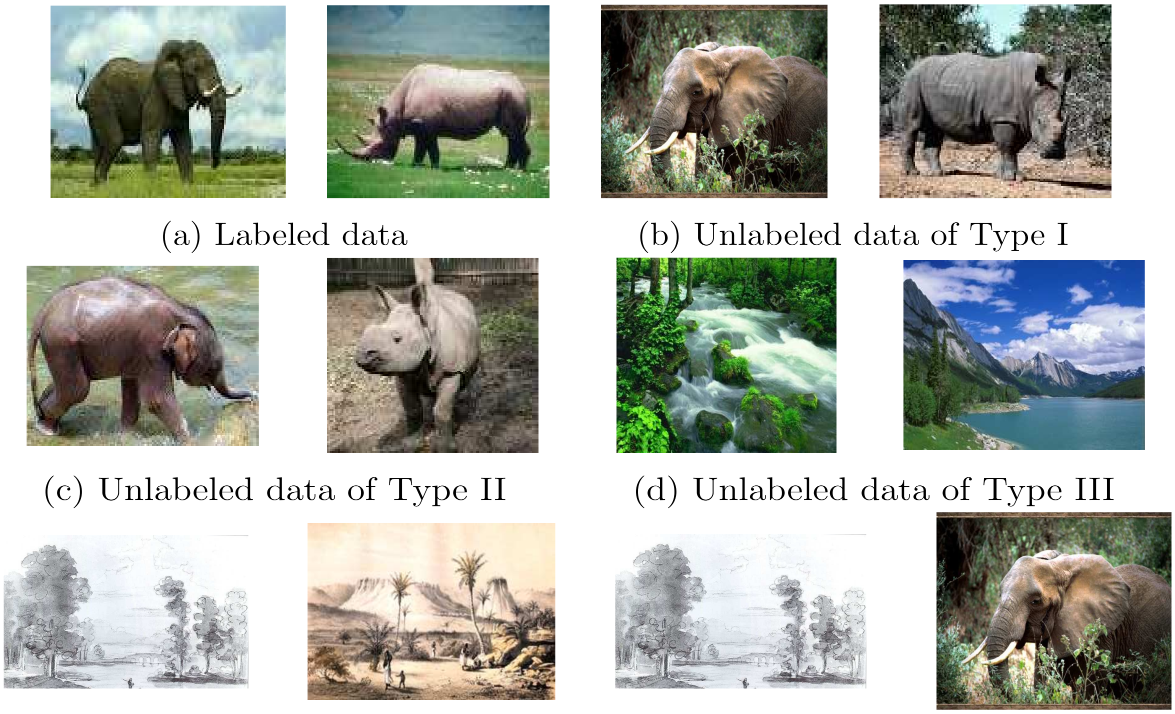
- Type II: the unlabeled data are drawn from a variance-drifted distribution and share the same labels with the training data. The representative learning framework is called learning under covariance shift or sample bias correction [123, 155]. For the above task, this type of unlabeled data can be illustrated as 2.1 (c), where the unlabeled images has a slightly different distribution as the labeled images in 2.1 (a). For example, the baby rhino has no teeth 2.1 (c), while the rhino in 2.1 (a) has.

- Type III: the unlabeled data share no common labels with the labeled data but are weakly-related to the labeled data only structurally. The resulting learning paradigm includes self-taught learning [112, 41]. For the above classification task, the unlabeled images could be pictures of scenery as shown in Figure 2.1 (d). The images in Figure 2.1 (d) may share similar high-level structure (such as textures) with images in Figure 2.1 (a).

- Type IV: irrelevant data or background data. The corresponding learning paradigm includes learning with universum [140]. An example of unlabeled images in this type could be shown in Figure 2.1 (e), where the images are not relevant to the classification task.

- Type V: mixture of two or three types of data. We call this learning paradigm as semi-supervised learning from a mixture [157, 63]. For the above classification task, the unlabeled images is a mixture of elephants, rhinos, and other irrelevant images, as shown in Figure 2.1 (f).

Each category of unlabeled data will lead to a kind of specific learning technique. Our review on these learning techniques will also be focused on this hierarchy structure. To make it clear, we also draw the relationship among these different types of unlabeled data and their derived learning approaches in Figure 2.2.



(a) Labeled data

(b) Unlabeled data of Type I

(c) Unlabeled data of Type II

(d) Unlabeled data of Type III

(e) Unlabeled data of Type IV

(f)Unlabeled data of Type V

Figure 2.1: The illustration of different types of unlabeled data.

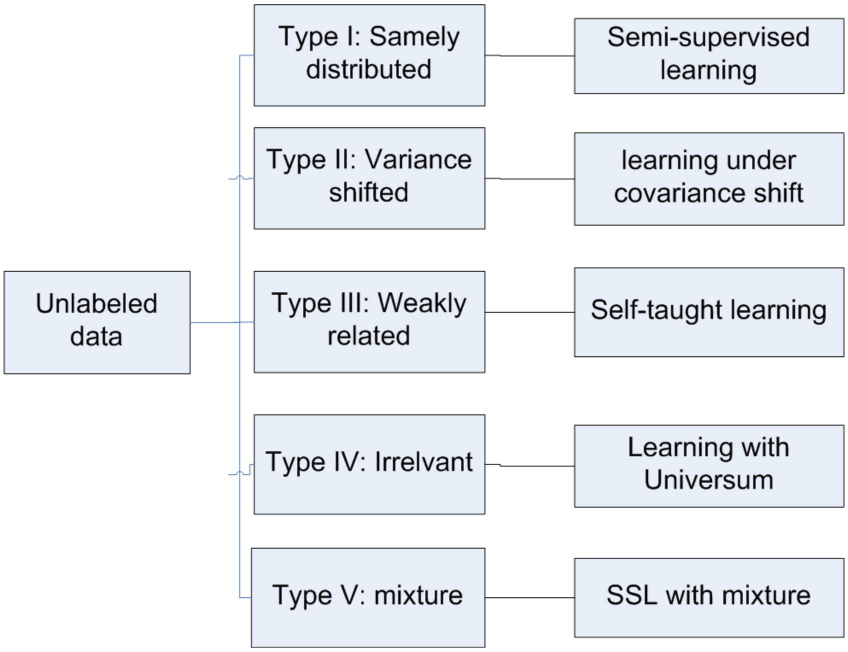


Figure 2.2: A categorization of unlabeled data and their related learning paradigms.

2.1 Supervised Learning

Statistical learning has achieved success in both the research and application areas [138]. Supervised statistical learning is a machine learning technique for learning a function from training data with the supervision of category information. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or a class label of the input pattern (called classification). There have been a lot of classification models motivated from different perspectives to solve the supervised classification problem. A number of statistical models have been proposed including Support Vector Machine (SVM) [130, 119, 139, 138], Fisher Discriminant Analysis (FDA) [51, 101, 69, 151], Logistic Regression [161], Gaussian Process classifiers [141], and Minimax Probability Machine

(MPM) [87, 88, 64, 67, 149, 68], etc. We briefly review SVM since it is currently regarded as the state-of-the-art classification model.

Among these models, Support Vector Machine (SVM) has attracted a lot of research focus, which improves the generalization ability by maximizing the margin between two different classes while keeping a small classification error. Theoretically, SVM is established to minimize the expected classification risk over the joint distribution $p(\mathbf{x}, \mathbf{y})$, which is defined as follows:

$$\mathcal{R}(\mathbf{f}) = \int_{\mathbf{z},\mathbf{y}} p(\mathbf{x}, \mathbf{y}) l(\mathbf{x}, \mathbf{y}, \mathbf{f}) \ , \tag{2.1}$$

where, $l(\mathbf{z}, \mathbf{y}, \mathbf{f})$ is the loss function. The above loss function describes the extent on how close the estimated class disagrees with the real class for the training data. Various metrics can be used for defining this loss function, among which the Hinge loss is the most used one.

As $p(\mathbf{x}, \mathbf{y})$ is usually unknown, people seek to approximate the above expected risk by the so-called empirical risk:

$$\mathcal{R}_{emp}(\mathbf{f}) = \frac{1}{n_l} \sum_{i=1}^{n_l} l(\mathbf{x}_i, y_i, \mathbf{f}) \ . \tag{2.2}$$

Since the empirical risk considers only the training data, it is easy to lead to the over-fitting problem. The Structure Risk Minimization principle [24, 139] is proposed instead in order to control the complexity of a learning function $\mathbf{f}$, which is also called VC dimension. Based on the Structure Risk Minimization principle, in SVM, the margin between two classes is maximized in order to reduce the over-fitting risk. A more formal explanation and theoretical foundation can be obtained from the Structure Risk Minimization criterion [24, 139]. Finally, SVM is defined as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n_l}\xi_i \qquad (2.3) \\ \text{s. t.} \quad & y_i(\mathbf{w}^\top\mathbf{x}_i - b) \geq 1 - \xi_i, \xi_i \geq 0,\ i = 1, 2, \ldots, n_l. \end{aligned}$$

where the decision function is defined as $f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} - b$, ξ_i is the margin error, and C is a pre-defined constant.

By involving a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{F}$, where $\mathcal{F}$ is the feature space, SVM can be represented as follows in the feature space:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n_l}\xi_i \qquad (2.4) \\ \text{s. t.} \quad & y_i(\mathbf{w}^\top\Phi(\mathbf{x}_i) - b) \geq 1 - \xi_i, \xi_i \geq 0,\ i = 1, 2, \ldots, n_l. \end{aligned}$$

According to the Lagrange techniques [138, 22], the above problem can be solved in the dual form:

$$\begin{aligned} \max_{\alpha} \quad & 2\sum_{i=1}^{n_l}\alpha_i - \sum_{i=1}^{n_l}\sum_{j=1}^{n_l}\alpha_i\alpha_j\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)y_iy_j \qquad (2.5) \\ \text{s. t.} \quad & \sum_{i=1}^{n_l}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C,\ i = 1, 2, \ldots, n_l. \end{aligned}$$

Using the kernel trick, one does not need to know the form of the mapping function. We import the following definition of a kernel. A kernel is a function κ, such that $\kappa(\mathbf{x}, \mathbf{z}) = \langle\Phi(\mathbf{x}), \Phi(\mathbf{z})\rangle$ for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$, where $\langle\cdot, \cdot\rangle$ is the operator of inner product. A kernel matrix is a square matrix $\mathbf{K} \in \mathcal{R}^{n\times n}$ such that $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ for some $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ and some kernel function κ.

We illustrate SVM in Figure 2.3 where two classes of data are depicted as circles and solid dots. Intuitively, there are many decision hyperplanes, which can be adopted for separating these

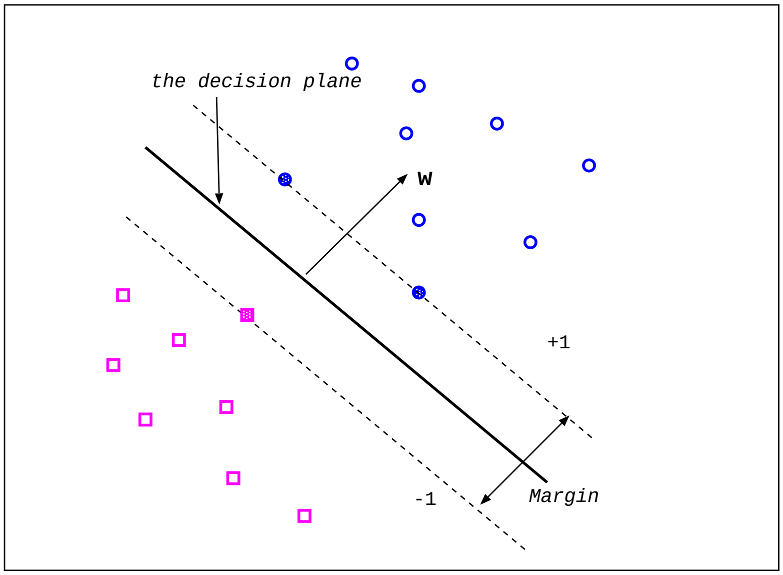


Figure 2.3: An illustration of Support Vector Machine

two classes of data. However, the one plotted in this figure is selected as the favorable separating plane, because it maximizes the margin between two classes. Therefore, in the objective function of SVM, a regularization term representing the margin shows up. Moreover, as seen in this figure, only those filled points, called support vectors, mainly determine the separating plane, while other points do not contribute to the margin at all.

2.2 Unsupervised Learning

Unsupervised learning is another type of machine learning where the labels of input data are not used during the learning process. It is distinguished from supervised learning where the learning process is supervised by the category or output information of training data. One of the most popular learning problems in unsupervised learning is data clustering [47, 71], which categorizes objects into different groups, or more precisely, partitions a data set into subsets (clusters), so that the data in the same subset are closer and the data among different subsets are further, according to some predefined distance measure. The data clustering problem has been studied for many years and many algorithms have been proposed. A recent survey on data clustering can be found in [146].

2.3 Semi-supervised Learning

Semi-supervised learning has attracted an increasing amount of research interest recently [29, 163]. Many semi-supervised learning models have proposed, including EM with generative mixture models [106], self-training, co-training, transductive support vector machines, and graph-based methods.

Generally, semi-supervised learning methods are derived based on two fundamental geometric assumptions: the low density assumption (or cluster assumption) and the manifold assumption. Before presenting the definitions of these two assumptions, we first give an illustration of them in Figure 2.4. In Figure 2.4 (a), the symbols $\oplus$ and $\ominus$ represent labeled data for the positive class and the negative class, respectively. The solid circles $\bullet$ represent unlabeled data. The dashed lines are the decision boundary lines obtained by the traditional supervised SVM that is built using labeled data. The solid lines are the decision boundary lines obtained by Transductive SVM that is constructed using both the labeled and unlabeled data. In Figure 2.4 (b), the red diamond and the blue circle represent labeled data for the positive class and the negative class, respectively. The unlabeled data are noted by the squares. The decision boundary curve are obtained by Laplancian SVM with manifold regularization. The left figure shows an example implementing low density assumption. It can be observed that when the unlabeled data are available, the decision boundary hyperplane represented by the solid lines are more reasonable. In the right figure, the decision boundary of Laplacian SVM, which is a semi-supervised learning method based on manifold assumption, are plotted. The decision boundary respects the manifold within the unlabeled data.

In the low density assumption, decision boundaries should not cross high density regions, but instead lie in low density regions. Semi-supervised learning methods based on this assumption usually minimize the losses on both the labeled and unlabeled data. Typical methods include label-switching-retraining [75], replacing the loss functions [37, 34], and convex relaxation methods [145, 148].

In the manifold assumption, data are assumed to form a low-dimensional manifold in some input space. Many semi-supervised methods implement such an assumption by using the graph Laplacian of a graph-based representation to characterize the manifold structure. Typical methods include semi-supervised spectral kernel learning [165], semi-supervised learning using gaussian fields [164], the point-cloud kernel [125], learning with local and global consistency [160], manifold regularization [10], etc. The relationship among them can be illustrated in Figure 2.5.

Recently, the relation between the low density assumption and the manifold assumption catches the attention of researchers in semi-supervised learning. More recently, [103] shows that the cut-size of the graph partition converges to the weighted volume of the boundary separating the two regions of the domain for a fixed partition. This takes a step toward the connection between graph-based partitioning to ideas surrounding low density assumption. However, current work cannot generalize the result uniformly over all partitions, therefore there is much work to do in order to build such a connection between graph-based partitioning to ideas surrounding low density assumption. [84]

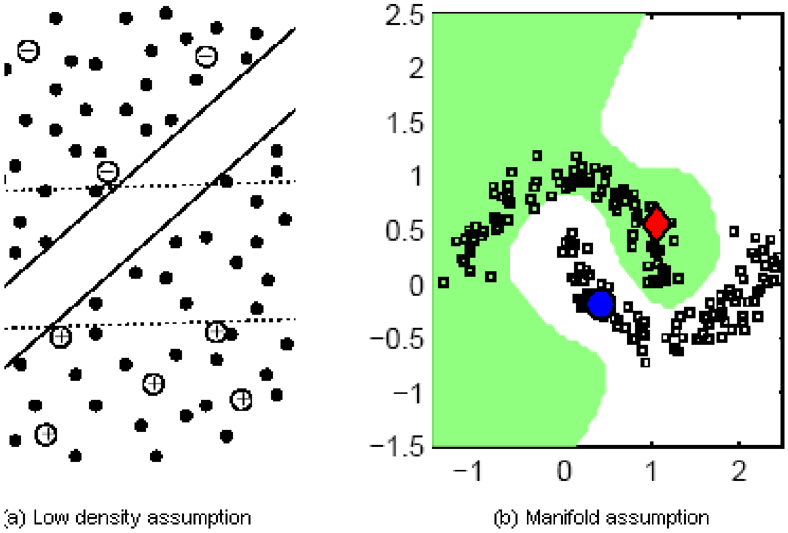


Figure 2.4: The illustration of two semi-supervised learning methods based on low density assumption and manifold assumption, respectively.

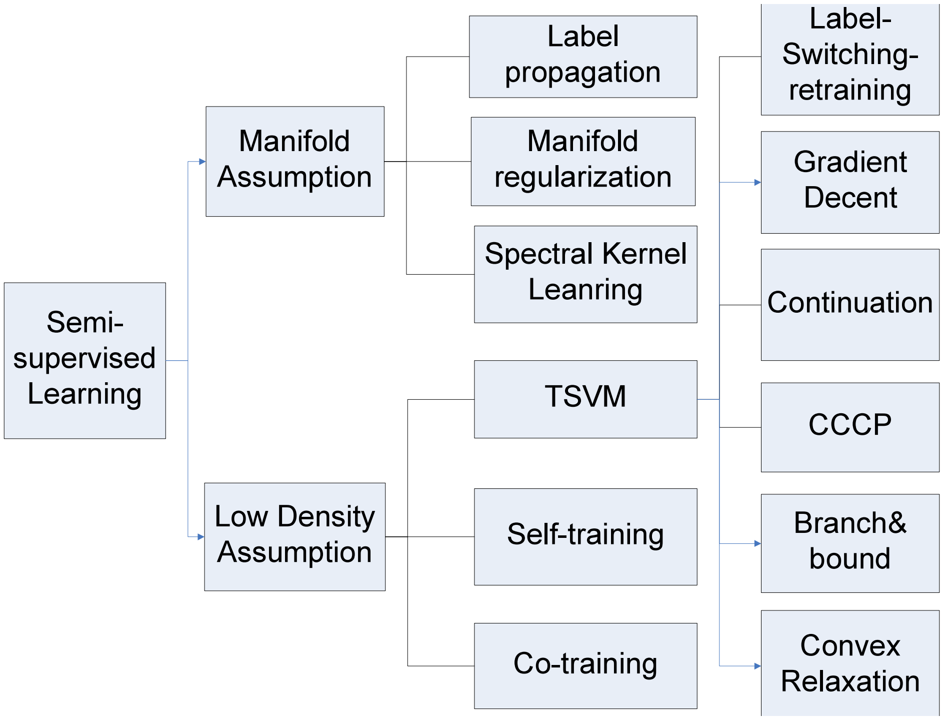


Figure 2.5: The relationship among semi-supervised learning, low density assumption, manifold assumption and their motivated methods.

studies the assumptions of semi-supervised learning from the viewpoint of minimax theory and suggests that decoupling the manifold assumption and the low density assumption is crucial to clarifying the problem.

2.3.1 Transductive Support Vector Machine

An important semi-supervised learning paradigm is the Transductive Support Vector Machine (TSVM), which maximizes the margin in the presence of unlabeled data and keeps the boundary traversing through low density regions, while respecting labels in the input space.

Since TSVM requires solving a combinatorial optimization problem, extensive research efforts have been devoted to efficiently finding the approximate solution to TSVM. The popular version of TSVM proposed in [75] uses a label-switching-retraining procedure to speed up the computation. In [34], the hinge loss in TSVM is replaced by a smooth loss function, and a gradient descent method is used to find the decision boundary in a region of low density. Chapelle et al. [27] employ an iterative approach for TSVM. It begins with minimizing an easy convex object function, and then gradually approximates the objective of TSVM with more complicated functions. The solution of the simple function is used as the initialization for the solution to the complicated function. Other iterative methods, such as deterministic annealing [124] and the concave-convex procedure (CCCP) method [37], are also employed to solve the optimization problem related to TSVM. The main drawback of the approximation methods listed above is that they are susceptible to local optima, and therefore are sensitive to the initialization of solutions. To address this problem, in [30], a branch-and-bound search method is developed to find the exact solution. In [145], the authors approximate TSVM by a semi-definite programming problem, which leads to a relaxation solution to TSVM (noted as RTSVM), to avoid the solution of local optimum. However, both approaches suffer from the high computational cost and can only be applied to small sized data sets. A more recent review on semi-supervised SVM can be found in [31].

2.3.2 Graph-based Semi-supervised Learning Models

Graph-based semi-supervised methods are another popular paradigms in semi-supervised learning. They usually define a graph where labeled and unlabeled examples form the nodes, and the similarity of examples are used to define edges. These methods usually assume label smoothness over the graph. Graph methods are nonparametric, discriminative, and transductive in nature.

Many graph-based methods work by estimating a function f over the graph, such that f satisfies two properties: (1) it should be close to the given labels of the labeled nodes, and (2) it should be smooth on the whole graph. This can be expressed in a regularization framework where the first term is a loss function, and the second term is a regularizer. The typical models in this category include the mincut [17], spectral graph transducer [76], Tikhonov regularization algorithm [121], and the local and global consistency method [160], the manifold regularization method [125, 10], and Gaussian Random Fields [164].

2.3.3 Other Semi-supervised Models

In this section, we simply introduce other semi-supervised learning models, which include EM with generative mixture models, self-training and co-training.

One example of EM with generative mixture models is proposed in [106] where the EM algorithm on mixture of multinomial is applied in the task of text classification. They showed the resulting classifiers perform better than those trained only from labeled data.

In self-training a classifier is first trained with the small amount of labeled data. The classifier is then used to classify the unlabeled data. Typically the most confident unlabeled points, together with their predicted labels, are added to the training set. Self-training has been applied to several natural language processing tasks. Yarowsky [153] uses self-training for word sense disambiguation. Co-training [19] assumes that there are two independent sets of features, each of which is sufficient to train a good classifier. Given the class, the two sets are conditionally independent. Initially two separate classifiers are trained with the labeled data on each feature subset. Each classifier then classifies the unlabeled data, and teaches the other classifier with the few unlabeled examples (and the predicted labels) that they feel most confident. Each classifier is retrained with the additional training examples given by the other classifier, and the process repeats. Co-training [7, 94] can be quiet effective, so that in the extreme case only one labeled point is needed to learn the classifier.

2.4 Learning from Variance-shifted Unlabeled Data

Different from the above semi-supervised learning paradigm that training data and test data are assumed to have the same distribution, another learning paradigm relaxes a little bit more than the above assumption: there is sample selection bias for the training sample and test sample. This problem is also called learning under covariance shift or sample bias correction [123, 155]. Other research efforts addressing on correcting sample selection bias include [49, 61]. More recently, [15] proposes a discriminative method dealing with the difference between training and test distributions and achieves success in the spam filtering domain.

2.5 Learning from Weakly-related Data

We describe the weakly-related data as those share structural information with the labeled data of the target domain. The weakly-related data therefore may not share the same labels as data in the target domain. The resulted learning is often called transfer learning [25]. In particular, [112] names the transfer learning from unlabeled data as self-taught learning.

Transfer learning, or Inductive Transfer, is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. Recently, transfer learning has been recognized as an important topic in machine learning research. Several researchers have proposed new approaches to solve the problems of transfer learning. Early transfer learning work raised some important issues, such as learning how to learn [118], learning one more thing [135], and multi-task learning [25]. A related topic is multi-task learning whose objective is to discover the common knowledge in multiple tasks. This common knowledge belongs to almost all the tasks, and is helpful for solving a new task. [11] provided a theoretical justification for multi-task learning. [43, 44] have studied the domain-transfer problem in statistical natural language processing, using a specific Gaussian model. [42] develop a boosting algorithm of transfer classification framework under the PAC learning model. [114] constructs informative priors using transfer learning.

Researchers also consider the problem of learning with auxiliary data, where a group of labeled data are treated as the auxiliary data. In previous work, [143] proposed an image classification algorithm using both inadequate training data and plenty of low quality auxiliary data. They demonstrated some improvement by using the auxiliary data. However, they did not give a quantitative study using different auxiliary examples. [95] improved learning with auxiliary data using active learning. [116] proposed a hierarchical Naive Bayes approach for transfer learning using auxiliary data, and discussed when transfer learning would improve the performance and when decrease.

When even the unlabeled same-class data are hard to obtain, one can also try some structurally-related unlabeled data. This is verified in [112], where the authors proposed a Self-taught Learning (STL) and showed that weakly-related unlabeled data sharing a little structural information with the current task could also benefit the classification performance. The problem is that those weakly-related data are only exploited for extracting feature patterns and they are not involved in optimizing the decision boundary. An empirical evaluation shows that self-taught learning sometimes extracts misleading patterns and hence might hurt the performance.

2.6 Learning with Universum

Another special kind of data is called universum [140], which does not belong to any classes of the problem at hand. [140] has shown that the universum data could boost the classification performance by encoding the prior knowledge of the domain.

In addition, [79] and [157] studied the case that unlabeled data are a mixture of both relevant data, which are from the same domain as the current task, and irrelevant data, which are from a different task or the background. More specifically, [157] assumed that the prior knowledge about the composition of the mixture, i.e., the universum data and the good quality same-domain data, is clear before learning a semi-supervised classification model.

2.7 Kernel Learning

Kernel methods have been playing an important role in statistical machine learning [119]. Kernel learning [86, 109, 100, 136] aims to learn a better pair-wised similarity measure and has received a lot of attention in recent studies of machine learning. It works by embedding the data from the input space to a Hilbert space, and then searching for relations among the embedded data points. The embedding implicitly defines the geometry of the feature space and induces a notion of similarity in the input space.

In this section, we briefly review recent work on kernel learning. First we review one of the most popular kernel learning methods which is called multiple kernel learning. Then we discuss semi-supervised kernel learning where the unlabeled data are utilized to assist the learning of kernel similarities.

2.7.1 Supervised Kernel Learning

A generic approach to learning a kernel function is known as multiple kernel learning (MKL) [86]: given a list of base kernel functions/matrices, MKL searches for the linear combination of base kernel functions which maximizes a generalized performance measure. Previous studies [86, 166, 154, 39, 4] have shown that MKL is usually able to identify appropriate combination of kernel functions, and as a result to improve the performance. Recent studies in bioinformatics also revealed the outstanding performance of multiple kernel learning in biological sequence classification and genomic data fusion[85, 131].

A variety of methods have been used to create base kernels. For instance, base kernels can be created by using different kernel functions; they can also be created by using a single kernel function but with different subsets of features. As for the performance measures needed to find the optimal kernel function, several measures have been studied for multiple kernel learning, including maximum margin classification errors [5, 86], exponential loss or logarithmic loss [38, 14], kernel-target alignment [39], Fisher discriminative analysis [80, 154], and cross-validation risk [32].

2.7.2 Semi-supervised Kernel Learning

Semi-supervised kernel learning approaches can usually be divided into two groups: the group based on spectral graph theory [36, 127], and the group based on multiple kernel learning or kernel selection [86].

In the first group, several semi-supervised learning algorithms have been proposed based on spectral graph theory, for example, cluster kernel [33], diffusion kernels [81], Gaussian fields [164], heat kernel [83], and the order-constrained spectral kernel [165]. Typically, a graph is constructed where the nodes are the data instances and the edges define the "local similarity" measures among data points. For example, the local similarity measure can be the Euclidean distance and the edge can be constructed by the node's k nearest neighbors. The edge between two data points suggests that they may share the same label. In general, it is believed that smaller eigenvalues correspond to smoother eigenvectors over the graph. Thus smaller eigenvalues and corresponding eigenvectors are used to compose the initial graph Laplacian which is further employed to maximize the alignment between the learned kernel matrix and the target kernel in order to learn a new kernel matrix. In [165], the experimental results imply that the order-constrained spectral kernel achieves better performance than the diffusion kernel and the Gaussian field kernel. Moreover, [59, 152] extend the spectral kernel learning method by specifying a fast spectral decay rate [59].

Some recent theoretical work builds the connection between spectral graph theory and kernel learning. Smola and Condor show some theoretical understanding between kernel and regularization based on the graph theory [127]. In addition, Berkin et al. develop a regularization framework for regularization on graphs [9]. Recently, Zhang et al. provide a theoretical framework for semi-supervised learning based on unsupervised kernel design and derive a generalization error bound [159]. It is demonstrated that a kernel with a fast decay rate is useful for the classification task [142, 159].

In the second group, [86] firstly extends multiple kernel learning from the supervised case to the transductive case by incorporating the whole part of kernel matrix into the learning process. Another way of learning a semi-supervised kernel function is combining graph Laplacians [3] since the pseudo inverse of the graph Laplacian can be regarded as a kernel. More recently, [40] proposes a kernel selection method for semi-supervised kernel machines, which can work for both the maximum-margin-based and manifold-regularization-based semi-supervised learning methods.

2.8 Semi-supervised Text Categorization

In this section, we review the related work on text categorization. As we discuss the situation where unlabeled data are not available, we also review work on query generation in order to actively download unlabeled Web pages from the Internet.

2.8.1 Related Work on Text Categorization

Text categorization is an active research topic in the communities of Web data mining, information retrieval, and statistical machine learning.

In the past decade, statistical learning techniques have been widely applied to text categorization [120], e.g, Bayesian classifiers [19], Support Vector Machines (SVM) [74], Logistic regression [158], and others. Empirical studies in recent years [74] have shown that SVM is the state-of-the-art technique.

Traditional text categorization is conducted in the supervised setting, namely learning a classification model for text categorization from a pool of labeled documents. The supervised setting often requires a large amount of labeled documents before a reliable classification model can be built. Hence, an important research question in text categorization is how to build reliable text classifiers given a limited number of labeled documents. The key is to effectively explore unlabeled documents for text categorization. The first approach toward semi-supervised text categorization is multi-view learning. The main idea to represent each document by multiple views and exploit unlabeled documents through the correlation among different views. This approach is especially effective for Web page and scientific document classification, in which the hyper-links between Web pages and the citation among research articles provide an additional representation for documents besides their textual contents [19, 122, 150]. Another example of multi-view learning is email categorization, in which the summaries of email texts [93] can be used as a complementary representation for emails. The co-training algorithm [19] and the EM algorithm for semi-supervised text categorization [106] also belong to this category. The second approach exploring unlabeled documents is to develop semi-supervised learning techniques that learn a classification model for text categorization from a mixture of labeled and unlabeled documents. The well-known examples within this category include Transductive SVM for text categorization [75, 148]. The third approach is active learning [97, 58, 59] that aims to choose the most informative unlabeled documents for manually labeling. Finally, in addition to semi-supervised learning and active learning, another approach toward text categorization with small-size samples is to transfer the knowledge of a related text categorization task to the target text categorization task, which is closely related to transfer learning [25], domain adaptation [43], or transfer leaning from weakly-related unlabeled documents [112, 52].

2.8.2 Related Work on Query Generation

Query generation is an important technique in information retrieval and natural language processing [6]. It is widely used to generate a corpus or expand an existing corpus for a given concept (see for example, [53, 20, 56, 23]). It is also used for generating topic hierarchies in question answering (see for example, [35]). Many query generation methods work by selecting a few representative words or key words from a small collection of documents or text segments based on certain statistical measures. The generated queries are then submitted to a Web search engine to retrieve a large set of related documents or text segments. In [20], a system is constructed using Web search agents to generate new queries and to extract the documents that are closely related to the given set of documents. In [53], a textual corpus of minority languages is constructed by querying Web search engines with the boolean queries that are constructed by the operator of conjunction and negation. In [77], the authors introduced and studied the problem of query substitution whose goal is to generate a new query that is closely related to a user's original search query. In [50], the authors modified the user's query by appending the keywords that are extracted from the SVM model for text categorization. Another example of query expansion can be found in [96] where the original user-input query is appended with additional terms relevant to some specific scenarios. In [26], the authors construct query concepts by clustering the features extracted from documents; however, the clustering accuracy needs to be supported by a large amount of training documents.

A number of statistical measures are used by the previous studies of query generation, including Odds-ratio [53], Term Frequency (TF) [53], Term Frequency/Inverse Document Frequency (TF/IDF) [20], and SVM-based measures [50, 89]. Although we can directly use the existing statistical measurement for query generation of the proposed text categorization framework, in this thesis we focus on the problem of query generation with a small number of documents. This is particularly challenging since most of the statistical measurements mentioned above cannot be estimated reliably when the number of labeled documents is small. Indeed, we will show in our empirical study that several statistical measurements proposed in the previous studies failed to identify the Web documents that are relevant to the target topics.

In the next chapter, we will derive an efficient convex relaxation model for Transductive SVM followed by extensive experimental evaluation.

□ **End of chapter.**

Chapter 3

Efficient Convex Relaxation for TSVM

Semi-supervised learning has attracted an increasing amount of research interest recently [29, 163]. An important semi-supervised learning paradigm is the Transductive Support Vector Machine (TSVM), which maximizes the margin in the presence of unlabeled data and keeps the boundary traversing through low density regions, while respecting labels in the input space.

Since TSVM requires solving a combinatorial optimization problem, extensive research efforts have been devoted to efficiently finding the approximate solution to TSVM. The popular version of TSVM proposed in [75] uses a label-switching-retraining procedure to speed up the computation. In [34], the hinge loss in TSVM is replaced by a smooth loss function, and a gradient descent method is used to find the decision boundary in a region of low density. Chapelle et al. [27] employ an iterative approach for TSVM. It begins with minimizing an easy convex object function, and then gradually approximates the objective of TSVM with more complicated functions. The solution of the simple function is used as the initialization for the solution to the complicated function. Other iterative methods, such as deterministic annealing [124] and the concave-convex procedure (CCCP) method [37], are also employed to solve the optimization problem related to TSVM. The main drawback of the approximation methods listed above is that they are susceptible to local optima, and therefore are sensitive to the initialization of solutions. To address this problem, in [30], a branch-and-bound search method is developed to find the exact solution. In [145], the authors approximate TSVM by a semi-definite programming problem, which leads to a relaxation solution to TSVM (noted as RTSVM), to avoid the solution of local optimum. However, both approaches suffer from the high computational cost and can only be applied to small sized data sets.

To this end, we present the convex relaxation for Transductive SVM (**CTSVM**). The key idea of our method is to approximate the non-convex optimization problem of TSVM by its dual problem. The advantage of doing so is twofold:

- Unlike the semi-definite relaxation [145] that approximates TSVM by dropping the rank constraint, the proposed approach approximates TSVM by its dual problem. As the basic result of convex analysis, the conjugate of conjugate of any function $f(\mathbf{x})$ is the convex envelope of $f(\mathbf{x})$, and therefore provides a tighter convex relaxation for $f(\mathbf{x})$ [57]. Hence, the proposed approach provides a better convex relaxation than that in [145] for the optimization problem in TSVM.

- Compared to the semi-definite relaxation TSVM, the proposed algorithm involves fewer free parameters and therefore significantly improves the efficiency by reducing the worst-case computational complexity from $\mathcal{O}(n^{6.5})$ to $\mathcal{O}(n^{4.5})$.

In the following section, we present the formulation of semi-definite relaxation for TSVM. Section 3.2 presents the proposed efficient convex relaxation approach for Transductive SVM. Section 3.3 presents the empirical studies that verify the effectiveness of the proposed relaxation for TSVM. Section 3.4 sets out the conclusion.

3.1 Convex Relaxation of TSVM

In this section, we review the key formulae for Transductive SVM, followed by the semi-definite programming relaxation for TSVM.

Let $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denote the entire data set, including both the labeled examples and the unlabeled ones. We assume that the first l examples within $\mathcal{X}$ are labeled by $\mathbf{y}_\ell = (y_1^\ell, y_2^\ell, \ldots, y_l^\ell)$ where $y_i^\ell \in \{-1, +1\}$ represents the binary class label assigned to $\mathbf{x}_i$. We further denote by $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{-1, +1\}^n$ the binary class labels predicted for all the data points in $\mathcal{X}$. The goal of TSVM is to estimate $\mathbf{y}$ by using both the labeled examples and the unlabeled ones.

Following the framework of maximum margin, TSVM aims to identify the classification model that will result in the maximum classification margin for both labeled and unlabeled examples, which amounts to solving the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{w}, b, \mathbf{y} \in \{-1,+1\}^n, \varepsilon} \quad & \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} \varepsilon_i \\
\text{s. t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0,\ i = 1, 2, \ldots, n \\
& y_i = y_i^\ell,\ i = 1, 2, \ldots, l,
\end{aligned}$$

where $C \geq 0$ is the trade-off parameter between the complexity of function $\mathbf{w}$ and the margin errors. The prediction function can be formulated as $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} - b$.

Evidently, the above problem is a non-convex optimization problem due to the product term $y_i w_j$ in the constraint. In order to approximate the above problem into a convex programming problem, we first rewrite the above problem into the following form using the Lagrange Theorem:

$$\begin{array}{ll}\min\limits_{\nu,\mathbf{y}\in\{-1,+1\}^n,\delta,\lambda} & \frac{1}{2}(\mathbf{e}+\nu-\delta+\lambda\mathbf{y})^\top\mathcal{D}(\mathbf{y})\mathbf{K}^{-1}\mathcal{D}(\mathbf{y})(\mathbf{e}+\nu-\delta+\lambda\mathbf{y})+C\delta^\top\mathbf{e}\\ \text{s. t.} & \nu\geq 0, \quad \delta\geq 0, \quad y_i=y_i^\ell,\ i=1,2,\ldots,l,\end{array}$$

where ν, δ and λ are the dual variables. $\mathbf{e}$ is the n-dimensional column vector of all ones and $\mathbf{K}$ is the kernel matrix. $\mathcal{D}(\mathbf{y})$ represents a diagonal matrix whose diagonal elements form the vector $\mathbf{y}$. Detailed derivation can be found in [86, 137]. Using the Schur complement, the above formulation can be further arranged as follows:

$$\begin{array}{ll}\min\limits_{\mathbf{y}\in\{-1,+1\}^n,t,\nu,\delta,\lambda} & t\\ \text{s. t.} & \begin{pmatrix}\mathbf{y}\mathbf{y}^\top\circ\mathbf{K} & \mathbf{e}+\nu-\delta+\lambda\mathbf{y}\\ (\mathbf{e}+\nu-\delta+\lambda\mathbf{y})^\top & t-2C\delta^\top\mathbf{e}\end{pmatrix}\succeq 0\\ & \nu\geq 0,\ \delta\geq 0,\ y_i=y_i^\ell,\ i=1,2,\ldots,l,\end{array} \tag{3.1}$$

where the operator $\circ$ represents the element wise product.

To convert the above problem into a convex optimization problem, the key idea is to replace the quadratic term $\mathbf{y}\mathbf{y}^\top$ by a linear variable. Based on the result that the set $\mathcal{S}_a=\{\mathbf{M}=\mathbf{y}\mathbf{y}^\top|\mathbf{y}\in\{-1,+1\}^n\}$ is equivalent to the set $\mathcal{S}_b=\{\mathbf{M}|M_{i,i}=1,\mathrm{rank}(\mathbf{M})=1\}$, we can approximate the problem in (3.1) as follows:

$$\begin{array}{ll}\min\limits_{\mathbf{M},t,\nu,\delta,\lambda} & t\\ \text{s. t.} & \begin{pmatrix}\mathbf{M}\circ\mathbf{K} & \mathbf{e}+\nu-\delta\\ (\mathbf{e}+\nu-\delta)^\top & t-2C\delta^\top\mathbf{e}\end{pmatrix}\succeq 0\\ & \nu\geq 0,\ \delta\geq 0,\\ & \mathbf{M}\succeq 0,\ M_{i,i}=1,\ i=1,2,\ldots,n,\end{array} \tag{3.2}$$

where $M_{ij}=y_i^\ell y_j^\ell$ for $1\leq i,j\leq l$.

Note that the key differences between (3.1) and (3.2) are (a) the rank constraint rank($\mathbf{M}$) = 1 is removed, and (b) the variable λ is set to be zero, which is equivalent to setting $b = 0$. The above approximation is often referred to as the Semi-Definite Programming (SDP) relaxation. As revealed by the previous studies [145, 16], the SDP programming problem resulting from the approximation is computationally expensive. More specifically, there are $\mathcal{O}(n^2)$ parameters in the SDP cone and $\mathcal{O}(n)$ linear inequality constraints, which implies a worst-case computational complexity of $\mathcal{O}(n^{6.5})$. To avoid the high computational complexity, we present a different approach for relaxing TSVM into a convex problem. Compared to the SDP relaxation approach, it is advantageous in that (1) it produces the best convex approximation for TSVM, and (2) it is computationally more efficient than the previous SDP relaxation.

3.2 Efficient Convex Relaxed Transductive Support Vector Machine

In this section, we follow the work of generalized maximum margin clustering [137] by first studying the case of hard margin, and then extending it to the case of soft margin.

3.2.1 Hard Margin TSVM

In the hard margin case, SVM does not penalize the classification error, which corresponds to $\delta = 0$ in (3.1). The resulting formulism of TSVM becomes

$$\begin{aligned}\min_{\nu,\mathbf{y},\lambda} \quad & \frac{1}{2}(\mathbf{e}+\nu+\lambda\mathbf{y})^\top \mathcal{D}(\mathbf{y})\mathbf{K}^{-1}\mathcal{D}(\mathbf{y})(\mathbf{e}+\nu+\lambda\mathbf{y}) \\ s.\ t. \quad & \nu \geq 0, \\ & y_i = y_i^\ell,\ i=1,2,\ldots,l, \\ & y_i^2 = 1,\ i=l+1,l+2,\ldots,n.\end{aligned} \tag{3.3}$$

Instead of employing the SDP relaxation as in [145], we follow the work in [137] and introduce a variable $\mathbf{z} = \mathcal{D}(\mathbf{y})(\mathbf{e}+\nu) = \mathbf{y}\circ(\mathbf{e}+\nu)$. Given that $\nu \geq 0$, the constraints in (3.3) can be written as $y_i^\ell z_i \geq 1$ for the labeled examples, and $z_i^2 \geq 1$ for all the unlabeled examples. Hence, $\mathbf{z}$ can be used as the prediction function, i.e., $f^* = \mathbf{z}$. Using this new notation, the optimization problem in (3.3) can be rewritten as follows:

$$\begin{aligned}\min_{\mathbf{z},\lambda} \quad & \frac{1}{2}(\mathbf{z}+\lambda\mathbf{e})^\top \mathbf{K}^{-1}(\mathbf{z}+\lambda\mathbf{e}) \\ \text{s. t.} \quad & y_i^\ell z_i \geq 1,\ i=1,2,\ldots,l, \\ & z_i^2 \geq 1,\ i=l+1,l+2,\ldots,n.\end{aligned} \tag{3.4}$$

One problem with Transductive SVMs is that it is possible to classify all the unlabeled data to one of the classes with a very large margin due to the high dimension and few labeled data. This will lead to poor generalization ability. To solve this problem, we introduce the following balance constraint to ensure that no class takes all the unlabeled examples:

$$-\epsilon \leq \frac{1}{l}\sum_{i=1}^{l} z_i - \frac{1}{n-l}\sum_{i=l+1}^{n} z_i \leq \epsilon, \tag{3.5}$$

where $\epsilon \geq 0$ is a constant. Through the above constraint, we aim to ensure that the difference between the labeled data and the unlabeled data in their class assignment is small.

To simplify the expression, we further define $\mathbf{w} = (\mathbf{z},\lambda) \in \mathbb{R}^{n+1}$ and $\mathbf{P} = (\mathbf{I}_n, \mathbf{e}) \in \mathbb{R}^{n\times(n+1)}$. Then, the problem in (3.4) becomes:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P} \mathbf{w} \\ \text{s. t.} \quad & y_i^\ell w_i \geq 1,\ i = 1, 2, \ldots, l, \\ & w_i^2 \geq 1,\ i = l+1, l+2, \ldots, n, \\ & -\epsilon \leq \frac{1}{l} \sum_{i=1}^{l} w_i - \frac{1}{n-l} \sum_{i=l+1}^{n} w_i \leq \epsilon. \end{aligned} \tag{3.6}$$

When this problem is solved, the label vector $\mathbf{y}$ can be directly determined by the sign of the prediction function, i.e., sign($\mathbf{w}$). This is because $w_i = (1+\nu)y_i$ for $i = l+1, \ldots, n$ and $\nu \geq 0$.

The following theorem shows that the problem in (3.6) can be relaxed to a semi-definite programming.

Theorem 1. *Given a sample $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and a partial set of the labels $\mathbf{y}_\ell = (y_1^\ell, y_2^\ell, \ldots, y_l^\ell)$ where $1 \leq l \leq n$, the variable $\mathbf{w}$ that optimizes (3.6) can be calculated by*

$$\mathbf{w} = \frac{1}{2} \left[\mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b})\right]^{-1} (\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}), \tag{3.7}$$

where $\mathbf{a} = (\mathbf{y}^l, \mathbf{0}^{n-l}, 0) \in \mathbb{R}^{n+1}$, $\mathbf{b} = (\mathbf{0}^l, \mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$, $\mathbf{c} = (\frac{1}{l}\mathbf{1}^l, -\frac{1}{u}\mathbf{1}^{n-l}, 0) \in \mathbb{R}^{n+1}$, $\mathbf{A} = \mathbf{P}^\top \mathbf{K}^{-1} \mathbf{P}$, and γ is determined by the following semi-definite programming:

$$\begin{aligned} \max_{\gamma, t, \alpha, \beta} \quad & -\frac{1}{4} t + \sum_{i=1}^{n} \gamma_i - \epsilon(\alpha + \beta) \\ s.\ t. \quad & \begin{pmatrix} \mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b}) & \gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}, \\ (\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c})^\top & t \end{pmatrix} \succeq 0 \\ & \alpha \geq 0,\ \beta \geq 0,\ \gamma_i \geq 0,\ i = 1, 2, \ldots, n. \end{aligned} \tag{3.8}$$

Proof Sketch. We define the Lagrangian of the minimization problem (3.6) as follows:

$$\begin{aligned}\min_{\mathbf{w}} \ \max_{\gamma} \ \mathcal{F}(\mathbf{w},\gamma) &= \mathbf{w}^\top \mathbf{P}^\top \mathbf{K}^{-1}\mathbf{P}\mathbf{w} + \sum_{i=1}^{l} \gamma_i(1 - y_i^\ell w_i) + \sum_{i=l+1}^{n} \gamma_i(1 - w_i^2) \\ &+\alpha(\mathbf{c}^\top \mathbf{w} - \epsilon) + \beta(-\mathbf{c}^\top \mathbf{w} - \epsilon),\end{aligned}$$

where $\gamma_i \geq 0$ for $i = 1, \ldots, n$. It can be derived from the duality that $\min_{\mathbf{w}} \ \max_{\gamma} \ \mathcal{F}(\mathbf{w},\gamma) = \max_{\gamma} \ \min_{\mathbf{w}} \ \mathcal{F}(\mathbf{w},\gamma)$.

At the optimum, the derivatives of $\mathcal{F}$ with respect to the variable $\mathbf{w}$ are derived as below:

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{w}} &= 2\left[\mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b})\right]\mathbf{w} - \gamma \circ \mathbf{a} \\ &+(\alpha - \beta)\mathbf{c} = 0,\end{aligned}$$

where $\mathbf{A} = \mathbf{P}^\top \mathbf{K}^{-1}\mathbf{P}$. The inverse of $\mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b})$ can be computed through adding a regularization parameter. Therefore, $\mathbf{w}$ is able to be calculated by:

$$\mathbf{w} = \frac{1}{2}\left[\mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b})\right]^{-1}(\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}).$$

Thus, the dual form of the problem becomes:

$$\begin{aligned}\max_{\gamma} \quad \mathcal{L}(\gamma) &= -\frac{1}{4}(\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c})^\top \left[\mathbf{A} - \mathcal{D}(b \circ \gamma)\right]^{-1}(\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}) \\ &+\sum_{i=1}^{n} \gamma_i - \epsilon(\alpha + \beta),\end{aligned}$$

We import a variable t, so that

$$-\frac{1}{4}(\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c})^\top [\mathbf{A} - \mathcal{D}(\mathbf{b} \circ \gamma)]^{-1}(\gamma \circ \mathbf{a} - (\alpha - \beta)\mathbf{c}) \geq -t.$$

According to the Schur Complement, we obtain a semi-definite programming cone, from which the optimization problem (3.8) can be formulated. ■

Remark I. The problem in (3.8) is a convex optimization problem, more specifically, a semi-definite programming problem, and can be efficiently solved by the interior-point method [105] implemented in some optimization packages, such as SeDuMi [133]. Besides, our relaxation algorithm has $\mathcal{O}(n)$ parameters in the SDP cone and $\mathcal{O}(n)$ linear equality constraints, which involves a worst-case computational complexity of $\mathcal{O}(n^{4.5})$. However, in the previous relaxation algorithms [16, 145], there are approximately $\mathcal{O}(n^2)$ parameters in the SDP cone, which involves a worst-case computational complexity in the scale of $\mathcal{O}(n^{6.5})$. Therefore, our proposed convex relaxation algorithm is more efficient. In addition, as analyzed in Section 3.1, the approximation in [16, 145] drops the rank constraint of the matrix $\mathbf{y}^\top\mathbf{y}$, which does not lead to a tight approximation. On the other hand, our prediction function f^* implements the conjugate of conjugate of the prediction function $f(\mathbf{x})$, which is the convex envelope of $f(\mathbf{x})$ [57]. Thus, our proposed convex approximation method provides a tighter approximation than the previous method.

Remark II. It is interesting to discuss the connection between the solution of the proposed algorithm and that of harmonic functions. We consider a special case of (3.7), where $\lambda = 0$ (which implies no bias term in the primal SVM), and there is no balance constraint. Then the solution of (3.8) can be expressed as follows:

$$\mathbf{z} = \frac{1}{2}\left[\mathbf{K}^{-1} - \mathcal{D}(\gamma \circ (\mathbf{0}^l, \mathbf{1}^{n-l}))\right]^{-1}(\gamma \circ (\mathbf{y}^l, \mathbf{0}^{n-l})). \tag{3.9}$$

It can be further derived as follows:

$$\mathbf{z} = \left(\mathbf{I}_n - \sum_{i=l+1}^{n} \gamma_i \mathbf{K}\mathbf{I}_n^i\right)^{-1} \left(\sum_{i=1}^{l} \gamma_i y_i^\ell \mathbf{K}(\mathbf{x}_i, \cdot)\right), \tag{3.10}$$

where $\mathbf{I}_n^i$ is defined as an $n \times n$ matrix with all elements being zero except the i-th diagonal element which is 1, and $\mathbf{K}(\mathbf{x}_i, \cdot)$ is the i-th column of $\mathbf{K}$. Similar to the solution of the harmonic function, we first propagate the class labels from the labeled examples to the unlabeled one by term $\sum_{i=1}^{l} \gamma_i y_i^\ell \mathbf{K}(\mathbf{x}_i, \cdot)$, and then adjust the prediction labels by the factor $\left(\mathbf{I}_n - \sum_{i=l+1}^{n} \gamma_i \mathbf{K}\mathbf{I}_n^i\right)^{-1}$. The key difference in our solution is that (1) different weights (i.e., γ_i) are assigned to the labeled examples, and (2) the adjustment factor is different from that in the harmonic function [164].

3.2.2 Soft Margin TSVM

We extend TSVM to the case of soft margin by considering the following problem:

$$\begin{aligned}
\min_{\nu,\mathbf{y},\delta,\lambda} \quad & \frac{1}{2}(\mathbf{e}+\nu-\delta+\lambda\mathbf{y})^\top \mathcal{D}(\mathbf{y})\mathbf{K}^{-1}\mathcal{D}(\mathbf{y})(\mathbf{e}+\nu-\delta+\lambda\mathbf{y}) \\
& + C_\ell \sum_{i=1}^{l} \delta_i^2 + C_u \sum_{i=l+1}^{n} \delta_i^2 \\
\text{s. t.} \quad & \nu \geq 0, \ \delta \geq 0, \\
& y_i = y_i^\ell, \ 1 \leq i \leq l, \\
& y_i^2 = 1, \ l+1 \leq i \leq n,
\end{aligned}$$

where δ_i is related to the margin error. Note that we distinguish the labeled examples from the unlabeled examples by introducing different penalty constants for margin errors: C_ℓ for labeled examples and C_u for unlabeled examples.

Similarly, we introduce the slack variable $\mathbf{z}$, and then derive the following dual problem:

$$\begin{aligned}
\max_{\gamma,t,\alpha,\beta} \quad & -\frac{1}{4}t + \sum_{i=1}^{n} \gamma_i - \epsilon(\alpha+\beta) \qquad (3.11)\\
s.\ t. \quad & \begin{pmatrix} \mathbf{A} - \mathcal{D}(\gamma \circ \mathbf{b}) & \gamma \circ \mathbf{a} - (\alpha-\beta)\mathbf{c} \\ (\gamma \circ \mathbf{a} - (\alpha-\beta)\mathbf{c})^\top & t \end{pmatrix} \succeq 0,\\
& 0 \le \gamma_i \le C_\ell,\ i = 1, 2, \ldots, l,\\
& 0 \le \gamma_i \le C_u,\ i = l+1, l+2, \ldots, n,\\
& \alpha \ge 0,\ \beta \ge 0,
\end{aligned}$$

which is also a semi-definite programming problem and can be solved similarly.

3.3 Experiments

In this section, we report an empirical study of the proposed method on several benchmark data sets.

3.3.1 Data Sets Description

To make evaluations comprehensive, we have collected four UCI data sets and three text data sets as our experimental testbeds. The UCI data sets include Iono, sonar, Banana, and Breast, which are widely used in data classification. The WinMac data set consists of the classes, mswindows and mac, of the Newsgroup20 data set. The IBM data set contains the classes, IBM and non-IBM, of the Newsgroup20 data set. The course data set is made of the course pages and non-course pages of the WebKb corpus. For each text data set, we randomly sample the data with the sample size of 60, 300 and 1000, respectively. Each resulted sample is noted by the suffix, “-s”, “-m”, or “-l” depending on whether the sample size is small, medium or large. Table 8.1 describes the information of these data sets, where d

represents the data dimensionality, l means the number of labeled data points, and n denotes the total number of examples.

Table 3.1: Data sets used in the experiments, where d represents the data dimensionality, l means the number of labeled data points, and n denotes the total number of examples.

Data set	d	l	n	**Data set**	d	l	n
Iono	34	20	351	WinMac-m	7511	20	300
Sonar	60	20	208	IBM-m	11960	20	300
Banana	4	20	400	Course-m	1800	20	300
Breast	9	20	300	WinMac-l	7511	50	1000
IBM-s	11960	10	60	IBM-l	11960	50	1000
Course-s	1800	10	60	Course-l	1800	50	1000

3.3.2 Experimental Protocol

To evaluate the efficiency of the proposed convex relaxation for TSVM, we compare the running time of the proposed CTSVM with that of the original semi-definite relaxation approach (RTSVM). Figure 3.1 shows the running time of both the semi-definite relaxation of TSVM in [145] and the proposed convex relaxation for TSVM versus increasing number of unlabeled examples. The data set used in this example is the Course data set (see the experiment section), and the number of labeled examples is 20. We clearly see that the proposed convex relaxation approach is considerably more efficient than the semi-definition approach. This is because compared to the semi-definite relaxation TSVM, the proposed algorithm involves fewer free parameters and therefore significantly improves the efficiency by reducing the worst-case computational complexity from $\mathcal{O}(n^{6.5})$ to $\mathcal{O}(n^{4.5})$.

To evaluate the effectiveness of the proposed CTSVM method, we choose the conventional SVM as our baseline method. In our experiments, we also make comparisons with three state-of-the-

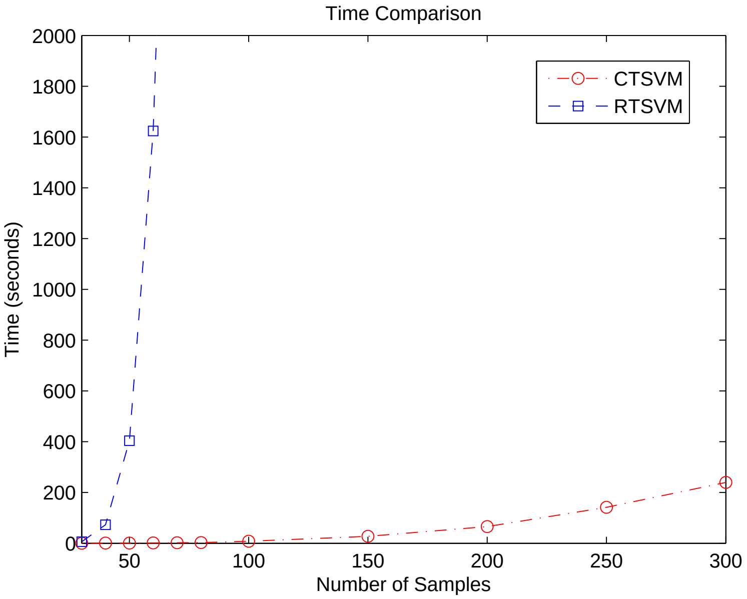


Figure 3.1: Computation time of the proposed convex relaxation approach for TSVM (i.e., CTSVM) and the semi-definite relaxation approach for TSVM (i.e., RTSVM) versus the number of unlabeled examples. The Course data set is used, and the number of labeled examples is 20.

art methods: the SVM-light algorithm [75], the Gradient Decent TSVM (∇TSVM) algorithm [34], and the Concave Convex Procedure (CCCP) [37]. Since the SDP approximation TSVM [145] has very high time complexity $O(n^{6.5})$, which is difficult to process data sets with hundreds of examples, it is only evaluated on the smaller data sets, i.e., “IBM-s” and “Course-s”.

The experimental setup is described as follows. For each data set, we conduct 10 trials. In each trial, the training set contains each class of data, and the remaining data are then used as the unlabeled (test) data. Moreover, the RBF kernel is used for “Iono”, “Sonar” and “Banana”, and the linear kernel is used for the other data sets. This is because the linear kernel performs better than the RBF kernel on these data sets. The kernel width of RBF kernel is chosen by 5-cross validation on the labeled data. The margin parameter C_ℓ is tuned by using the labeled data in all algorithms. Due to the small number of labeled examples, for CTSVM and CCCP, the margin parameter for unlabeled data, C_u, is set equal to C_ℓ. Other parameters in these algorithms are set to the default values according to the relevant literatures.

3.3.3 Experimental Results

Table 3.2: The classification performance of Transductive SVMs on benchmark data sets.

Data Set	**SVM**	**SVM-light**	**∇TSVM**	**CCCP**	**CTSVM**
Iono	78.55±4.83	78.25±0.36	81.72±4.50	**82.11**±3.83	80.09±2.63
Sonar	51.76±5.05	55.26±5.88	**69.36**±4.69	56.01±6.70	67.39±6.26
Banana	58.45±7.15	-	71.54±7.28	79.33±4.22	**79.51**±3.02
Breast	96.46±1.18	95.68±1.82	97.17±0.35	96.89±0.67	**97.79**±0.23
IBM-s	52.75±15.01	67.60±9.29	65.80±6.56	65.62±14.83	**75.25**±7.49
Course-s	63.52±5.82	76.82±4.78	75.80±12.87	74.20±11.50	**79.75**±8.45
WinMac-m	57.64±9.58	79.42±4.60	81.03±8.23	84.28±8.84	**84.82**±2.12
IBM-m	53.00±6.83	67.55±6.74	64.65±13.38	69.62±11.03	**73.17**±0.89
Course-m	80.18±1.27	**93.89**±1.49	90.35±3.59	88.78±2.87	92.92±2.28
WinMac-l	60.86±10.10	89.81±2.10	90.19±2.65	91.00±2.42	**91.25**±2.67
IBM-l	61.82±7.26	**75.40**±2.26	73.11±1.99	74.80±1.87	73.42±3.23
Course-l	83.56±3.10	92.35±3.02	93.58±2.68	91.32±4.08	**94.62**±0.97

Table 3.2 summarizes the classification accuracy and the standard deviations of the proposed algorithm, the baseline method and the state-of-the-art methods. It can be observed that our proposed algorithm performs significantly better than the standard SVM across all the data sets. Moreover, on the small-size data sets, i.e., “IBM-s” and “Course-s”, the results of the SDP-relaxation method are 68.57±22.73 and 64.03±7.65, respectively, which are worse than the proposed CTSVM method. In addition, the proposed CTSVM algorithm performs much better than other TSVM methods over “WinMac-m” and “Course-

l". As shown in Table 3.2, the SVM-light algorithm achieves the best results on "Course-m" and "IBM-l"; however, it fails to converge on "Banana". On the remaining data sets, comparable results have been obtained for our proposed algorithm. From the above, the empirical evaluations indicate that our proposed CTSVM method achieves promising classification results comparing with the state-of-the-art methods.

3.4 Summary and Future Work

This chapter presents a novel method for Transductive SVM by relaxing the unknown labels to the continuous variables. In contrast to the previous relaxation method which involves $\mathcal{O}(n^2)$ free parameters in the semi-definite matrix, our method takes the advantages of reducing the number of free parameters to $\mathcal{O}(n)$, and can solve the optimization problem more efficiently. In addition, the proposed approach provides a tighter convex relaxation for the optimization problem in TSVM. Empirical studies on benchmark data sets demonstrate that the proposed method is more efficient than the previous semi-definite relaxation method and achieves promising classification results comparing to the state-of-the-art methods.

As the current model is only designed for a binary-classification, we plan to develop a multi-class Transductive SVM model in the future. Moreover, it is desirable to extend the current model to classify the new incoming data which are not seen during the training process.

□ **End of chapter.**

Chapter 4

Level Method for Efficient Multiple Kernel Learning

The multiple kernel learning (MKL) problem was first formulated as a semi-definite programming (SDP) problem by [86]. An SMO-like algorithm was proposed in [5] in order to solve medium-scale problems. More recently, a Semi-Infinite Linear Programming (SILP) approach was developed for MKL [132]. SILP is an iterative algorithm that alternates between the optimization of kernel weights and the optimization of the SVM classifier. In each step, given the current solution of kernel weights, it solves a classical SVM with the combined kernel; it then constructs a cutting plane model for the objective function and updates the kernel weights by solving a corresponding linear programming problem. Although the SILP approach can be employed for large scale MKL problems, it often suffers from slow convergence [21]. One shortcoming of the SILP method is that it updates kernel weights solely based on the cutting plane model. Given that a cutting plane model usually differs significantly from the original objective function when the solution is far away from the points where the cutting plane model is constructed, the optimal solution to the cutting plane model could be significantly off target. In [115], the authors addressed the MKL problems by a simple Subgradient Descent (SD) method.

However, since the SD method is memoryless, it does not utilize the gradients computed in previous iterations, which could be very useful in boosting the efficiency of the search.

To further improve the computational efficiency of MKL, we extended the level method [92], which was originally designed for optimizing non-smooth functions, to the optimization of convex-concave problems. In particular, we regard the MKL problem as a saddle point problem. In the present work, similar to the SILP method, we construct in each iteration a cutting plane model for the target objective function using the solutions to the intermediate SVM problems. A new solution for kernel weights is obtained by solving the cutting plane model. We furthermore adjust the new solution via a projection to a level set. This adjustment is critical in that it ensures on one hand the new solution is sufficiently close to the current solution, and on the other hand the new solution significantly reduces the objective function. We show that the extended level method has a convergence rate of $\mathcal{O}(1/\varepsilon^2)$ for a ε-accurate solution. Although this is similar to that of the SD method, the extended level method is advantageous in that it utilizes all the gradients that have been computed so far. Empirical results with eight UCI datasets show that the extended level method is able to greatly improve the efficiency of multiple kernel learning in comparison with the SILP method and the SD method. Morevoer, in order to extend the level method to semi-supervised kernel learning. We warp each base kernel with the point cloud norm [125] of the unlabeled data. In this way, we could select the best subset of kernels for semi-supervised learning. Experiments on the USPS data set indicate its promising effect.

In the remaining of this chapter, we first summarize the framework of multiple kernel learning, then describe the details of the extended level method for MKL, including a study of its convergence rate. Then, we present experimental results by comparing both the effectiveness and the efficiency of the extended level method with the corresponding measures of SILP and SD followed by the concluding remarks.

4.1 Multiple Kernel Learning

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{n \times d}$ denote the collection of n training samples that are in a d-dimensional space. We further denote by $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in \{-1, +1\}^n$ the binary class labels for the data points in $\mathbf{X}$. We employ the maximum margin classification error, an objective used in SVM, as the generalized performance measure. Following [86], the problem of multiple kernel learning for classification in the primal form is defined as follows:

$$\min_{\mathbf{p} \in \mathcal{P}} \max_{\alpha \in \mathcal{Q}} \; f(\mathbf{p}, \alpha) = \alpha^\top \mathbf{e} - \frac{1}{2}(\alpha \circ \mathbf{y})^\top \left(\sum_{i=1}^{m} p_i \mathbf{K}_i \right) (\alpha \circ \mathbf{y}), (4.1)$$

where $\mathcal{P} = \{\mathbf{p} \in \mathbb{R}^m : \mathbf{p}^\top \mathbf{e} = 1, \; 0 \leq \mathbf{p} \leq 1\}$ and $\mathcal{Q} = \{\alpha \in \mathbb{R}^n : \alpha^\top \mathbf{y} = 0, \; 0 \leq \alpha \leq C\}$ are two solid convex regions, denoting the set of kernel weights and the set of SVM dual variables, respectively. Here, $\mathbf{e}$ is a vector of all ones, C is the trade-off parameter in SVM, $\{\mathbf{K}_i\}_{i=1}^m$ is a group of base kernel matrices, and $\circ$ defines the element-wise product between two vectors. It is easy to verify that $f(\mathbf{p}, \alpha)$ is convex on $\mathbf{p}$ and concave on α. Thus the above optimization problem is indeed a convex-concave problem. It is important to note that the block-minimization formulation of MKL presented in [115, 5] is equivalent to (4.1).

A straightforward approach toward solving the convex-concave problem in (4.1) is to transform it into a Semi-definite Programming (SDP) or a Quadratically Constrained Quadratic Programming (QCQP) [86, 5]. However, given their computational complexity, they cannot be applied to large-scale MKL problems. Recently, Semi-infinite Linear Programming (SILP) [132] and Subgradient Descent (SD) [115] have been applied to handle large-scale MKL problems. We summarize them into a unified framework in Algorithm 1. Note that a superscript is used to indicate the index of iteration, a convention that is used throughout this chapter. We use $[x]^t$ to denote x to the power of t in the case of ambiguity.

Algorithm 1 A general framework for solving MKL

```
1: Initialize $\mathbf{p}^0 = \mathbf{e}/m$ and $i = 0$
2: repeat
3:     Solve the dual of SVM with kernel $\mathbf{K} = \sum_{j=1}^m p_j^i \mathbf{K}_j$ and obtain optimal
       solution $\alpha^i$
4:     Update kernel weights by $\mathbf{p}^{i+1} = \arg\min\{f^i(\mathbf{p};\alpha) : \mathbf{p} \in \mathcal{P}\}$
5:     Update $i = i + 1$ and calculate stopping criterion $\Delta^i$
6: until $\Delta^i \leq \varepsilon$
```

As indicated in Algorithm 1, both methods divide the MKL problem into two cycles: the inner cycle solves a standard SVM problem to update α, and the outer cycle updates the kernel weight vector $\mathbf{p}$. They differ in the 4th step in Algorithm 1: the SILP method updates $\mathbf{p}$ by solving a cutting plane model, while the SD method updates $\mathbf{p}$ using the subgradient of the current solution. More specifically, $\varphi^i(\mathbf{p};\alpha)$ for SILP and SD are defined as follows:

$$\varphi^i_{SILP}(\mathbf{p};\alpha) = \min_{\nu}\{\nu : \nu \geq f(\mathbf{p},\alpha^j),\ j = 0,\ldots,i\}, \quad (4.2)$$

$$\varphi^i_{SD}(\mathbf{p};\alpha) = \frac{1}{2}\|\mathbf{p} - \mathbf{p}^i\|_2^2 + \gamma_i(\mathbf{p} - \mathbf{p}^i)^\top \nabla_{\mathbf{p}} f(\mathbf{p}^i,\alpha^i), \quad (4.3)$$

where γ_i is the step size that needs to be decided dynamically (e.g., by a line search). $\nabla_{\mathbf{p}} f(\mathbf{p}^i,\alpha^i) = -\frac{1}{2}[(\alpha^i \circ \mathbf{y})^\top \mathbf{K}_1(\alpha^i \circ \mathbf{y}),\ldots,(\alpha^i \circ \mathbf{y})^\top \mathbf{K}_m(\alpha^i \circ \mathbf{y})]^\top$ denotes the subgradient of $f(\cdot,\cdot)$ with respect to $\mathbf{p}$ at $(\mathbf{p}^i,\alpha^i)$. Comparing the two methods, we observe

- In SILP, the cutting plane model $\varphi_{SILP}(\mathbf{p})$ utilizes all the $\{\alpha^j\}_{j=1}^i$ obtained in past iterations. In contrast, SD only utilizes α^i of the current solution $\mathbf{p}^i$.

- SILP updates the solution for $\mathbf{p}$ based on the cutting plane model $\varphi_{SILP}(\mathbf{p})$. Since the cutting plane model is usually inaccurate when $\mathbf{p}$ is far away from $\{\mathbf{p}^j\}_{j=1}^i$, the updated solution $\mathbf{p}$ could be significantly off target [21]. In contrast, a regularization term $\|\mathbf{p} - \mathbf{p}^i\|_2^2/2$ is introduced in SD to prevent the new solution from being far off the current one, $\mathbf{p}^i$.

The proposed level method combines the strengths of both methods. Similar to SILP, it utilizes the gradient information of all the iterations; similar to SD, a regularization scheme is introduced to prevent the updated solution from being too far from the current solution.

4.2 Extended Level Method for MKL

We first introduce the basic steps of the level method, followed by the extension of the level method to convex-concave problems and its application to MKL.

4.2.1 Introduction to the Level Method

The level method [92] is from the family of bundle methods, which have recently been employed to efficiently solve regularized risk minimization problems [129]. It is an iterative approach designed for optimizing a non-smooth objective function. Let $f(x)$ denote the convex objective function to be minimized over a convex domain G. In the ith iteration, the level method first constructs a lower bound for $f(x)$ by a cutting plane model, denoted by $g^i(x)$. The optimal solution, denoted by $\hat{x}^i$, that minimizes the cutting plane model $g^i(x)$ is then computed. An upper bound $\overline{f}^i$ and a lower bound $\underline{f}_i$ are computed for the optimal value of the target optimization problem based on $\hat{x}^i$. Next, a level set for the cutting plane model $g^i(x)$ is constructed, denoted by $\mathcal{L}^i = \{x \in G : g^i(x) \leq \lambda\overline{f}^i + (1-\lambda)\underline{f}^i\}$ where $\lambda \in (0,1)$ is a tradeoff constant. Finally, a new solution x^{i+1} is computed by projecting x^i onto the level set $\mathcal{L}^i$. It is important to note that the projection step, serving a similar purpose to the regularization term in SD, prevents the new solution x^{i+1} from being too far away from the old one x^i. To demonstrate this point, consider a simple example $\min_x\{f(x) = [x]^2 : x \in [-4, 4]\}$. Assume $x^0 = -3$ is the initial solution. The cutting plane model at x^0 is $g^0(x) = 9 - 6(x + 3)$. The optimal solution minimizing $g^0(x)$ is $\hat{x}^1 = 4$. If we directly take $\hat{x}^1$ as the new solution, as SILP does, we found it is significantly worse than x^0 in terms of $[x]^2$. The level method alleviates this problem by projecting $x^0 = -3$ to the level set $\mathcal{L}^0 = \{x : g^0(x) \leq 0.9[x^0]^2 + 0.1g^0(\hat{x}^1), -4 \leq x \leq 4\}$ where $\lambda = 0.9$. It is easy to verify that the projection of x^0 to $\mathcal{L}^0$ is $x^1 = -2.3$, which significantly reduces the objective function $f(x)$ compared with x^0. This is illustrated in Figure 4.1.

4.2.2 Extension of the Level Method to MKL

We now extend the level method, which was originally designed for optimizing non-smooth functions, to convex-concave optimization. Different from the traditional level method, the extended level method solves the convex-concave optimization by employing an alternative procedure as described in Algorithm 1.

First, since $f(\mathbf{p}, \alpha)$ is convex in $\mathbf{p}$ and concave in α, according to van Neuman Lemma, for any optimal solution $(\mathbf{p}^*, \alpha^*)$ we have

$$f(\mathbf{p}, \alpha^*) = \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha) \geq f(\mathbf{p}^*, \alpha^*) \geq f(\mathbf{p}^*, \alpha) = \min_{\mathbf{p} \in \mathcal{P}} f(\mathbf{p}, \alpha) \quad (4.4)$$

This observation motivates us to design an MKL algorithm which iteratively updates both the lower and the upper bounds for $f(\mathbf{p}, \alpha)$ in order to find the saddle point. To apply the level method, we first construct the cutting plane model. Let $\{\mathbf{p}^j\}_{j=1}^i$

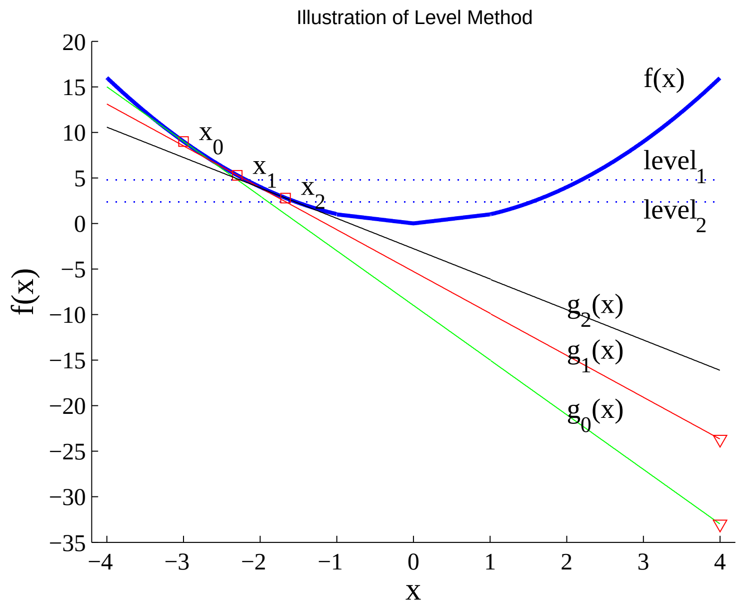


Figure 4.1: Illustration of the Level method. We aim to minimize $f(x)$ over [-4,4]. With the help of the affine lower bound function $g_i(x)$, we are able to gradually approximate the optimal solution. ∇ denotes the lower bound value in one iteration.

denote the solutions for $\mathbf{p}$ obtained in the last i iterations. Let $\alpha^j = \arg\max_{\alpha \in \mathcal{Q}} f(\mathbf{p}^j, \alpha)$ denote the optimal solution that maximizes $f(\mathbf{p}^j, \alpha)$. We construct a cutting plane model $g^i(\mathbf{p})$ as follows:

$$g^i(\mathbf{p}) = \max_{1 \le j \le i} f(\mathbf{p}, \alpha^j). \tag{4.5}$$

We have the following proposition for the cutting plane model $g^i(x)$.

Proposition 1. *For any* $\mathbf{p} \in \mathcal{P}$*, we have (a)* $g^{i+1}(\mathbf{p}) \ge g^i(\mathbf{p})$*, and (b)* $g^i(\mathbf{p}) \le \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha)$.

Next, we construct both the lower and the upper bounds for the optimal value $f(\mathbf{p}^*, \alpha^*)$. We define two quantities $\underline{f}^i$ and $\overline{f}^i$ as follows:

$$\underline{f}^i = \min_{\mathbf{p} \in \mathcal{P}} g^i(\mathbf{p}) \quad \textit{and} \quad \overline{f}^i = \min_{1 \le j \le i} f(\mathbf{p}^j, \alpha^j). \tag{4.6}$$

The following theorem shows that $\{\underline{f}^j\}_{j=1}^i$ and $\{\overline{f}^j\}_{j=1}^i$ provide a series of increasingly tight bounds for $f(\mathbf{p}^*, \alpha^*)$.

Theorem 1. *We have the following properties for* $\{\underline{f}^j\}_{j=1}^i$ *and* $\{\overline{f}^j\}_{j=1}^i$*: (a)* $\underline{f}^i \leq f(\mathbf{p}^*, \alpha^*) \leq \overline{f}^i$*, (b)* $\overline{f}^1 \geq \overline{f}^2 \geq \ldots \geq \overline{f}^i$*, and (c)* $\underline{f}^1 \leq \underline{f}^2 \leq \ldots \leq \underline{f}^i$*.*

Proof. First, since $g^i(\mathbf{p}) \leq \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha)$ for any $\mathbf{p} \in \mathcal{P}$, we have

$$\underline{f}^i = \min_{\mathbf{p} \in \mathcal{P}} g^i(\mathbf{p}) \leq \min_{\mathbf{p} \in \mathcal{P}} \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha).$$

Second, since $f(\mathbf{p}^j, \alpha^j) = \max_{\alpha \in Q} f(\mathbf{p}^j, \alpha)$, we have

$$\overline{f}^i = \min_{1 \leq j \leq i} f(\mathbf{p}^j, \alpha^j) = \min_{\mathbf{p} \in \{\mathbf{p}_1, \ldots, \mathbf{p}_i\}} \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha) \geq \min_{\mathbf{p} \in \mathcal{P}} \max_{\alpha \in \mathcal{Q}} f(\mathbf{p}, \alpha) = f(\mathbf{p}^*, \alpha^*).$$

Combining the above results, we have (a) in the theorem. It is easy to verify (b) and (c). □

We furthermore define the gap Δ^i as

$$\Delta^i = \overline{f}^i - \underline{f}^i.$$

The following corollary indicates that the gap Δ^i can be used to measure the sub-optimality for solution $\mathbf{p}^i$ and α^i.

Corollary 2. *(a)* $\Delta^j \geq 0, j = 1, \ldots, i$*, (b)* $\Delta^1 \geq \Delta^2 \geq \ldots \geq \Delta^i$*, (c)* $|f(\mathbf{p}^j, \alpha^j) - f(\mathbf{p}^*, \alpha^*)| \leq \Delta^i$

It is easy to verify these three properties of Δ^i in the above corollary using the results of Theorem 1.

In the third step, we construct the level set $\mathcal{L}^i$ using the estimated bounds $\overline{f}^i$ and $\underline{f}^i$ as follows:

$$\mathcal{L}^i = \{\mathbf{p} \in \mathcal{P} : g^i(\mathbf{p}) \leq \ell^i = \lambda \overline{f}^i + (1 - \lambda) \underline{f}^i\}, \qquad (4.7)$$

where $\lambda \in (0, 1)$ is a predefined constant. The new solution, denoted by $\mathbf{p}^{i+1}$, is computed as the projection of $\mathbf{p}^i$ onto the level set $\mathcal{L}^i$, which is equivalent to solving the following optimization problem:

$$\mathbf{p}^{i+1} = \arg\min_{\mathbf{p}} \left\{ \|\mathbf{p} - \mathbf{p}^i\|_2^2 : \mathbf{p} \in \mathcal{P}, f(\mathbf{p}, \alpha^j) \leq \ell^i, j = 1, \ldots, i \right\} \quad (4.8)$$

Although the projection is regarded as a quadratic programming problem, it can often be solved efficiently because its solution is likely to be the projection onto one of the hyperplanes of polyhedron $\mathcal{L}^i$. In other words, only very few linear constraints of $\mathcal{L}$ are active; most of them are inactive. This sparse nature usually leads to significant speedup of QP, similar to the solver of SVM. As we argue in the last subsection, by means of the projection, we on the one hand ensure $\mathbf{p}^{i+1}$ is not very far away from $\mathbf{p}^i$, and on the other hand ensure significant progress is made in terms of $g^i(\mathbf{p})$ when the solution is updated from $\mathbf{p}^i$ to $\mathbf{p}^{i+1}$. Note that the projection step in the level method saves the effort of searching for the optimal step size in SD, which is computationally expensive as will be revealed later. We summarize the steps of the extended level method in Algorithm 2.

Algorithm 2 The Level Method for Multiple Kernel Learning

```
Initialize $\mathbf{p}^0 = \mathbf{e}/m$ and $i = 0$
repeat
    Solve the dual problem of SVM with $\mathbf{K} = \sum_{j=1}^m p_j^i \mathbf{K}_j$ to obtain the optimal solution $\alpha^i$
    Construct the cutting plane model $g^i(\mathbf{p})$ in (4.5)
    Calculate the lower bound $\underline{f}^i$ and the upper bound $\overline{f}^i$ in (4.6), and the gap $\Delta^i$ in (4.2.2)
    Compute the projection of $\mathbf{p}^i$ onto the level set $\mathcal{L}^i$ by solving the optimization problem in (4.8)
    Update $i = i + 1$
until $\Delta^i \leq \varepsilon$
```

Finally, we discuss the convergence behavior of the level method. In general, convergence is guaranteed because the gap Δ^i, which bounds the absolute difference between $f(\mathbf{p}^*, \alpha^*)$ and $f(\mathbf{p}^i, \alpha^i)$, monotonically decreases through iterations. The following theorem shows the convergence rate of the level method when applied to multiple kernel learning.

Theorem 3. *To obtain a solution* $\mathbf{p}$ *that satisfies the stopping criterion, i.e.,*

$$|\max_{\alpha\in\mathcal{Q}} f(\mathbf{p},\alpha) - f(\mathbf{p}^*,\alpha^*)| \leq \varepsilon,$$

the maximum number of iterations N *that the level method requires is bounded as follows*

$$N \leq \frac{2c(\lambda)L^2}{\varepsilon^2}, \tag{4.9}$$

where

$$c(\lambda) = \frac{1}{(1-\lambda)^2\lambda(2-\lambda)} \quad \text{and} \quad L = \frac{1}{2}\sqrt{m}nC^2 \max_{1\leq i\leq m} \Lambda_{\max}(\mathbf{K}_i).$$

The operator $\Lambda_{\max}(M)$ *computes the maximum eigenvalue of matrix* M*.*

Due to space limitation, we put the proof of Theorem 3 into Appendix A. Theorem 3 tells us that the convergence rate of the level method is $\mathcal{O}(1/\varepsilon^2)$. It is important to note that according to Information Based Complexity (IBC) theory, given a function family $\mathcal{F}(L)$ with a fixed Lipschitz constant L, $\mathcal{O}(1/\varepsilon^2)$ is almost the optimal convergence rate that can be achieved for any optimization method based on the black box first order oracle. In other words, no matter which optimization method is used, there always exists a function $f(\cdot) \in \mathcal{F}(L)$ such that the convergence rate is $\mathcal{O}(1/\varepsilon^2)$ as long as the optimization method is based on a black box first order oracle. More details can be found in [104, 92].

4.2.3 Semi-supervised Multiple Kernel Learning

It is natural to extend the above multiple kernel learning approach to semi-supervised learning if each base kernel matrix incorporates the information within the unlabeled data. One way to incorporate the information of the unlabeled data is manifold regularization [121, 125]. Following this philosophy, one can estimation the geometry of the underlying marginal distribution from the unlabeled data. This geometry information can be further deformed into a new kernel matrix. The resulting new kernel can therefore be computed explicitly in terms of the unlabeled data. [125] has showed that working with only labeled data in this new RKHS, one can achieve competitive performance over semi-supervised methods.

It can be proved that the new kernel that warps the structure of the Reproducing Kernel Hilbert Space (RKHS) can be computed through deforming the original kernel function $\kappa(\mathbf{x}, \mathbf{z})$ using the graph laplacian $\mathcal{L}$:

$$\tilde{\kappa}(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{x}, \mathbf{z}) - \mathbf{k}_{\mathbf{x}}(\mathbf{I} + \mathcal{L}\mathbf{K})^{-1}\mathcal{L}\mathbf{k}_{\mathbf{z}}, \tag{4.10}$$

where $\tilde{\kappa}(\mathbf{x}, \mathbf{z})$ is the new kernel function that warps the information of the unlabeled data. $\mathbf{I}$ is the identity matrix. $\mathbf{k}_{\mathbf{x}}$ denotes the vector $(\kappa(\mathbf{x}_1, \mathbf{x}), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}))^T$. $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$.

According to [125], the original kernel matrix $\mathbf{K}$ plays an important role to form the new kernel matrix for semi-supervised learning. Usually, $\mathbf{K}$ can be constructed as an RBF kernel. However, it is difficult to select the kernel width for the kernel matrix $\mathbf{K}$, especially in the small sample learning problems. In order to solve this problem, we apply the multiple kernel learning approach to select the kernel width from a possible parameter set. Fortunately, given a set of candidate kernel widths, we could easily apply the level method to solve the multiple kernel learning problem for semi-supervised learning.

4.3 Experiments

We conduct experiments to evaluate the efficiency of the proposed algorithm for MKL in contrast with SILP and SD, the two state-of-the-art algorithms for MKL.

4.3.1 Experiments on supervised MKL

Experimental Setup

We follow the settings in [115] to construct the base kernel matrices, i.e.,

- Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \dots, 2^{6}\}$) on all features and on each single feature
- Polynomial kernels of degree 1 to 3 on all features and on each single feature.

Each base kernel matrix is normalized to unit trace. The experiments are conducted on a PC with 3.2GHz CPU and 2GB memory. According to the above scheme of constructing base kernel matrices, we select a batch of UCI data sets, with the cardinality and dimension allowed by the memory limit of the PC, from the UCI repository for evaluation. We repeat all the algorithms 20 times for each data set. In each run, 50% of the examples are randomly selected as the training data and the remaining data are used for testing. The training data are normalized to be zero mean and unit variance, and the test data are then normalized using the mean and variance of the training data. The regularization parameter C in SVM is set to 100 as our focus is to evaluate the computational time, as justified in [115]. For a fair comparison among the MKL algorithms, we adopt the same stopping criterion for all three algorithms under comparison: we adopt the duality gap criterion used in [115], i.e., $\max\limits_{1 \le i \le m} (\alpha \circ \mathbf{y})^\top \mathbf{K}_i (\alpha \circ \mathbf{y}) - (\alpha \circ \mathbf{y})^\top \left(\sum_{j=1}^{m} p_j \mathbf{K}_j \right) (\alpha \circ \mathbf{y})$, and stop the algorithm when the criterion is less than 0.01 or the number of iterations larger than 500. We empirically initialize the parameter λ to 0.9 and increase it to 0.99 when the ratio Δ_i/ℓ_i is less than 0.01 for all experiments, since a larger λ accelerates the projection when the solution is close to the optimal one [1]. We use the SimpleMKL toolbox [115] to implement the SILP and SD methods. The linear programming in the SILP method and the auxiliary subproblems in the level method are solved using a general optimization toolbox MOSEK (`http://www.mosek.com`).

Experimental Results

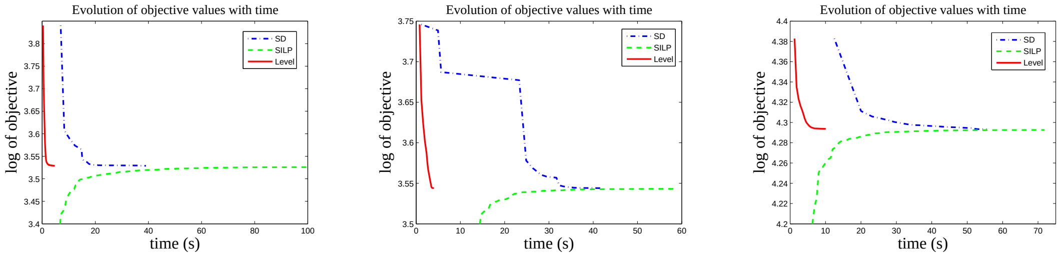
We report the following performance measures: prediction accuracy, training time, and the averaged number of kernels selected. From Table 4.1, we observe that all algorithms achieve almost the same prediction accuracy under the same stopping criterion. This is not surprising because all algorithms are essentially trying to solve the same optimization problem. Regarding the computational efficiency, we observe that the time cost of the SILP approach is the highest among all the three MKL algorithms. For datasets "Iono" and "Sonar", the SILP method consumes more than 30 times the computational cycles of the other two methods for MKL. We also observe that the level method is the most efficient among the three methods in comparison. To obtain a better picture of the computational efficiency of the proposed level method, we compute the time-saving ratio, as shown in Table 4.2. We observe that the level method saves 91.9% of computational time on average when compared with the SILP method, and 70.3% of computational time when compared with the SD method.

In order to see more details of each optimization algorithm,

[1] It is important to note that the convergence rate is still assured though the value of λ is changed during optimization. This can be easily proved since the algorithm will converge for any feasible value of λ.

we plot the logarithm values of the MKL objective function to base 10 against time in Figure 4.2. Due to space limitation, we randomly choose only three datasets, “Iono”, “Breast”, and “Pima”, as examples. It is interesting to find that the level method converges overwhelmingly faster than the other two methods. The efficiency of the level method arises from two aspects: (a) the cutting plane model utilizes the computational results of all iterations and therefore boosts the search efficiency, and (b) the projection to the level sets ensures the stability of the new solution. A detailed analysis of the SD method reveals that a large number of function evaluations are consumed in order to compute the optimal step size via a line search. Note that in convex-concave optimization, every function evaluation in the line search of SD requires solving an SVM problem. As an example, we found that for dataset “Iono”, although SD and the level method require similar numbers of iterations, SD calls the SVM solver 1231 times on average, while the level method only calls the solver 47 times. For the SILP method, the high computational cost is mainly due to the oscillation of the solutions. This instability leads to very slow convergence when the solution is close to the optimal one, as indicated by the long tail of SILP in Figure 4.2. The instability of SILP is further confirmed by the examination of kernel weights, as shown below.

To understand the evolution of kernel weights (i.e., $\mathbf{p}$), we plot the evolution curves of the five largest kernel weights for datasets “Iono”, “Breast”, and “Pima” in Figure 4.3. We observe that the values of $\mathbf{p}$ computed by the SILP method are the most unstable due to oscillation of the solutions to the cutting plane models. Although the unstable-solution problem is to some degree improved by the SD method, we still clearly observe that $\mathbf{p}$ fluctuates significantly through iterations. In contrast, for the proposed level method, the values of $\mathbf{p}$ change smoothly through iterations. We believe that the stability of the level method is mainly due to the accurate estimation of bounds as well as the regularization of the projection to the level sets. This observation also sheds light on why the level method can be more efficient than the SILP and the SD methods.



(a) *Iono* (b) *Breast* (c) *Pima*

Figure 4.2: Evolution of objective values over time (seconds) for datasets "Iono", "Breast", and "Pima". The objective values are plotted on a logarithm scale (base 10) for better comparison. Only parts of the evolution curves are plotted for SILP due to their long tails.

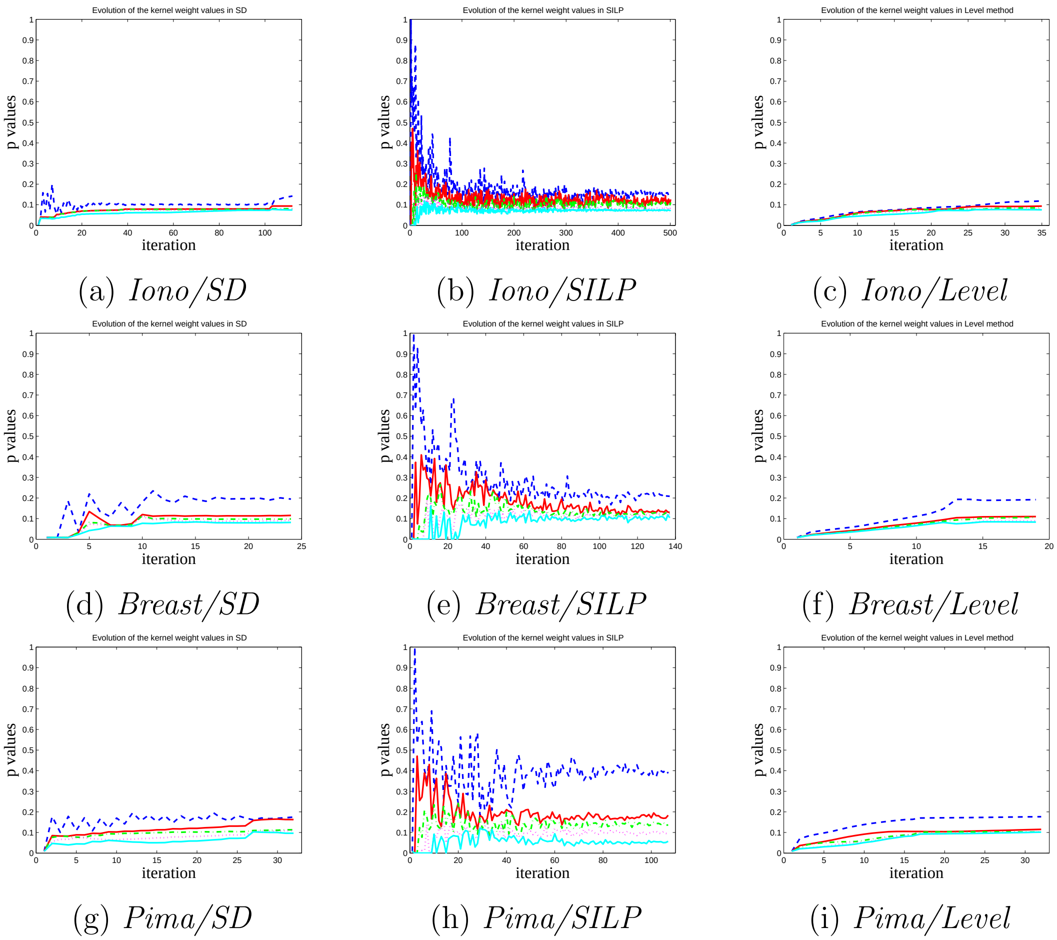
4.3.2 Experiments on Semi-supervised MKL

We use the USPS data set for testing the semi-supervised multiple kernel learning algorithm. USPS is a widely used data set for testing semi-supervised learning algorithms. We design five pairwise classification tasks (1 vs 7, 2 vs 7, 3 vs 8, 4 vs 7, 2 vs 3) of varying difficulty to test the performance of the semi-supervised kernel learning algorithm. For each task, we randomly select 400 data points to form the whole data.

We use a similar rule to construct the base kernel matrices for the deforming in the semi-supervised setting, i.e.,

- Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \ldots, 2^{6}\}$) on all features,
- Polynomial kernels of degree 1 to 3 on all features,
- linear kernel on each single feature.

The above setting is a little different from the supervised setting because of the constraint of the memory and the calculation of



(a) *Iono/SD* (b) *Iono/SILP* (c) *Iono/Level*

(d) *Breast/SD* (e) *Breast/SILP* (f) *Breast/Level*

(g) *Pima/SD* (h) *Pima/SILP* (i) *Pima/Level*

Figure 4.3: The evolution curves of the five largest kernel weights for datasets "Iono", "Breast" and "Pima" computed by the three MKL algorithms

the deformed kernel matrix $\tilde{\mathbf{K}}$.

We repeat all the algorithms 30 times for each data set. In each run, 5% of the examples are randomly selected as the training data and the remaining data are used as the unlabeled data. We follow the same data processing procedures as the supervised setting. Finally, we report the results on five tasks in Table 4.3. It is easy to verify that the extended level method is more efficient than the SILP method and the subgradient method as it is done in the supervised setting. It is also interesting to find that the level method achieves better classification accuracy as well. This indicates that the level method could be an efficient algorithm for semi-supervised multiple kernel learning.

4.4 Summary and Future Work

In this chapter, we propose an extended level method to efficiently solve the multiple kernel learning problem. In particular, the level method overcomes the drawbacks of both the SILP method and the SD method for MKL. Unlike the SD method that only utilizes the gradient information of the current solution, the level method utilizes the gradients of all the solutions that are obtained in past iterations; meanwhile, unlike the SILP method that updates the solution only based on the cutting plane model, the level method introduces a projection step to regularize the updated solution. It is the employment of the projection step that guarantees finding an updated solution that, on the one hand, is close to the existing one, and one the other hand, significantly reduces the objective function. Our experimental results have shown that the level method is able to greatly reduce the computational time of MKL over both the SD method and the SILP method in both supervised and semi-supervised settings.

For future work, we plan to find a scheme to adaptively set the value of λ in the level method and apply the level method to other tasks, such as one-class classification, multi-class classification, and regression.

□ **End of chapter.**

Table 4.1: The performance comparison of three MKL algorithms. Here n and m denote the size of training samples and the number of kernels, respectively.

	SD	**SILP**	**Level**
	Iono $n = 175$ $m = 442$		
Time(s)	33.5 ±11.6	1161.0 ±344.2	7.1 ±4.3
Accuracy (%)	92.1 ±2.0	92.0 ±1.9	92.1±1.9
#Kernel	26.9 ±4.0	24.4 ±3.4	25.4±3.9
	Breast $n = 342$ $m = 117$		
Time(s)	47.4 ±8.9	54.2 ±9.4	4.6 ±1.0
Accuracy (%)	96.6 ±0.9	96.6 ±0.8	96.6±0.8
#Kernel	13.1 ±1.7	10.6 ±1.1	13.3±1.5
	Pima $n = 384$ $m = 117$		
Time(s)	39.4 ±8.8	62.0 ±15.2	9.1 ±1.6
Accuracy (%)	76.9 ±1.9	76.9 ±2.1	76.9±2.1
#Kernel	16.6 ±2.2	12.0 ±1.8	17.6±2.6
	Sonar $n = 104$ $m = 793$		
Time(s)	60.1 ±29.6	1964.3±68.4	24.9±10.6
Accuracy (%)	79.1 ±4.5	79.3 ±4.2	79.0±4.7
#Kernel	39.8 ±3.9	34.2 ±2.6	38.6±4.1
	Wpbc $n = 198$ $m = 442$		
Time(s)	7.8 ±2.4	142.0 ±122.3	5.3 ±1.3
Accuracy (%)	77.0 ±2.9	76.9 ±2.8	76.9±2.9
#Kernel	19.5 ±2.8	17.2 ±2.2	20.3±2.6
	Heart $n = 135$ $m = 182$		
Time(s)	4.7 ±2.8	79.2 ±38.1	2.1 ±0.4
Accuracy (%)	82.2 ±2.2	82.2 ±2.0	82.2±2.1
#Kernel	17.5 ±1.8	15.2 ±1.5	18.6±1.9
	Vote $n = 218$ $m = 205$		
Time(s)	23.7 ±9.7	26.3 ±12.4	4.1 ±1.3
Accuracy (%)	95.7 ±1.0	95.7 ±1.0	95.7±1.0
#Kernel	14.0 ±3.6	10.6 ±2.6	13.8±2.6
	Wdbc $n = 285$ $m = 403$		
Time(s)	122.9±38.2	146.3 ±48.3	15.5±7.5
Accuracy (%)	96.7 ±0.8	96.5 ±0.9	96.7±0.8
#Kernel	16.6 ±3.2	12.9 ±2.3	15.6±3.0

Table 4.2: Time-saving ratio of the level method over the SILP and the SD method

	Iono	Breast	Pima	Sonar	Wpbc	Heart	Vote	Wdbc	Average
Level/SD (%)	78.9	90.4	77.0	58.7	32.5	54.7	82.8	87.4	70.3
Level/SILP (%)	99.4	91.6	85.4	98.7	88.7	97.3	84.5	89.4	91.9

Table 4.3: The performance comparison of three MKL algorithms for semi-supervised learning.

	SD	**SILP**	**Level**
	1 vs 7		
Time(s)	13.7±10.7	511.6±698.9	**2.7**±1.1
Accuracy (%)	96.2±4.1	94.6±9.1	**96.5** ±3.6
#Kernel	8.4±2.8	7.2±2.7	9.4±2.8
	2 vs 3		
Time(s)	17.0± 27.8	1362.0±611.4	**2.4**±1.4
Accuracy (%)	86.9±2.9	86.9±3.1	**87.2**±3.0
#Kernel	13.1±2.9	11.7±1.9	14.4±2.9
	2 vs 7		
Time(s)	16.3±10.5	1249.5±684.3	**2.5**±1.0
Accuracy (%)	88.3±3.9	88.1±4.0	**88.6**±3.8
#Kernel	12.4±2.4	10.2±1.9	13.4± 2.9
	3 vs 8		
Time(s)	11.6±9.8	990.0±726.1	**2.4**±1.3
Accuracy (%)	85.4±4.5	85.5±4.6	**85.8**±4.5
#Kernel	13.6±2.6	11.7±1.7	14.7±2.5
	4 vs 7		
Time(s)	13.6±9.2	671.8±682.2	**1.7**±0.7
Accuracy (%)	86.9±5.7	87.0±5.6	**87.2**±5.8
#Kernel	11.3±2.0	9.9±1.6	13.2±2.7

Chapter 5

Unified Framework for Semi-supervised Learning

Semi-supervised learning methods are derived from two fundamental geometric assumptions: the low density assumption (or cluster assumption) and the manifold assumption [84]. One typical semi-supervised learning model based on cluster assumption is Transductive SVM (TSVM) [75]; while one representative of methods based on manifold assumption is manifold regularization (also called Laplacian SVM) [10]. We discuss the relationship between Transductive SVM and the approach of manifold regularization in this chapter. Although these two types of approaches are based on different motivations, they essentially share similar spirit, namely the decision boundary should be decided by not only the labeled examples, but also the structure of the unlabeled examples. In the framework of transductive SVM, the regularization of decision boundary by the unlabeled data is achieved by the minimization of the loss function for the unlabeled data [75, 145, 148]. In contrast, the manifold regularization approach regulates the choice of decision boundary by an additional term of regularizer that is constructed by the Laplacian of the unlabeled data[125, 10]. In this chapter, we will first show that the unlabeled data used by TSVM can essentially be viewed as an additional regularizer for the decision boundary.

We then show that this additional regularizer induced by the TSVM is closely related to the regularizer introduced by the manifold regularization.

5.1 TSVM: A Regularization View

In this section, we will show that the role of unlabeled data within the framework of TSVM can also be viewed as an additional regularizer for the decision boundary.

Instead of using the original form of the TSVM, we introduce the following form of TSVM that can be derived through the duality

$$\begin{aligned} \min_{\mathbf{z},\delta} \quad & \frac{1}{2}\mathbf{z}^\top \mathbf{K}^{-1}\mathbf{z} + C\sum_{i=1}^{n_l}\delta_i \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l \\ & z_i^2 \geq 1,\ n_l+1 \leq i \leq n. \end{aligned} \tag{5.1}$$

In order to control the strength of the regularization produced by the unlabeled examples, we introduce the parameter $\rho \geq 0$ and modify the above problem (5.1) as :

$$\begin{aligned} \min_{\mathbf{z},\delta} \quad & \frac{1}{2}\mathbf{z}^\top \mathbf{K}^{-1}\mathbf{z} + C\sum_{i=1}^{n_l}\delta_i \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l \\ & z_i^2 \geq \rho,\ n_l+1 \leq i \leq n. \end{aligned} \tag{5.2}$$

Clearly, when $\rho = 1$, we have standard TSVM. In particular, the larger the ρ is, the stronger the regularization of the unlabeled data is. It is also important to note that in the above we only consider the classification error of the labeled examples, namely we only denote δ_i for each labeled example.

To facilitate the discussion, we write $\mathbf{z} = (\mathbf{z}_l; \mathbf{z}_u)$ where $\mathbf{z}_l = (z_1^l, \ldots, z_{n_l}^l)$ and $\mathbf{z}_u = (z_1^u, \ldots, z_{n_u}^u)$ represent the prediction for the labeled and the unlabeled examples, respectively. According to the inverse lemma of the block matrix, we can write $\mathbf{K}^{-1}$ as follows:

$$\mathbf{K}^{-1} = \left(\begin{array}{cc} \mathbf{C}_l^{-1} & -\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1} \\ -\mathbf{C}_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1} & \mathbf{C}_u^{-1} \end{array} \right)$$

where

$$\begin{aligned} \mathbf{C}_l &= \mathbf{K}_{l,l} - \mathbf{K}_{l,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{u,l} \\ \mathbf{C}_u &= \mathbf{K}_{u,u} - \mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u} \end{aligned}$$

Thus, the term $\mathbf{z}^\top\mathbf{K}^{-1}\mathbf{z}$ is computed as

$$\mathbf{z}^\top\mathbf{K}^{-1}\mathbf{z} = \mathbf{z}_l^\top\mathbf{C}_l^{-1}\mathbf{z}_l + \mathbf{z}_u^\top\mathbf{C}_u^{-1}\mathbf{z}_u - 2\mathbf{z}_l^\top\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{z}_u$$

Note that when the unlabeled data are loosely related to the labeled data, namely most of the elements within $\mathbf{K}_{u,l}$ are small, we will have $\mathbf{C}_u \approx \mathbf{K}_u$. We refer to this situation as "*weakly unsupervised learning*". Using the above equalities, we can rewrite TSVM as follows:

$$\begin{aligned} \min_{\mathbf{z}_l,\mathbf{z}_u,\delta} \quad & \frac{1}{2}\mathbf{z}_l^\top\mathbf{C}_l^{-1}\mathbf{z}_l + C\sum_{i=1}^{n_l}\delta_i + \omega(\mathbf{z}_l,\rho) \qquad (5.3) \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l, \end{aligned}$$

where $\omega(\mathbf{z}_l,\rho)$ is a regularization function for $\mathbf{z}_l$ and it is the output of the following optimization problem

$$\begin{aligned} \min_{\mathbf{z}_u} \quad & \frac{1}{2}\mathbf{z}_u^\top\mathbf{C}_u^{-1}\mathbf{z}_u - \mathbf{z}_l^\top\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{z}_u \qquad (5.4) \\ \text{s. t.} \quad & [z_i^u]^2 \geq \rho, \quad 1 \leq i \leq n_u \end{aligned}$$

To understand the regularization function $\omega(\mathbf{z}_l,\rho)$, we first compute the dual of the problem (5.4) by calculating the Lagrangian function:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\mathbf{z}_u^\top\mathbf{C}_u^{-1}\mathbf{z}_u - \mathbf{z}_l^\top\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{z}_u - \sum_{i=1}^{n_u}\frac{1}{2}\lambda_i([z_i^u]^2-\rho) \\ &= \frac{1}{2}\mathbf{z}_u^\top(\mathbf{C}_u^{-1} - D(\lambda))\mathbf{z}_u - \mathbf{z}_l^\top\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{z}_u + \rho\lambda^\top\mathbf{e}, \end{aligned}$$

where $D(\lambda) = \mathrm{diag}(\lambda_1, \ldots, \lambda_{n_u})$. By setting the derivative to be zero, we have

$$\begin{aligned}\mathbf{z}_u &= (\mathbf{C}_u^{-1} - D(\lambda))^{-1}\mathbf{C}_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ &= (\mathbf{I} - \mathbf{C}_u D(\lambda))^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l\end{aligned}$$

The dual problem becomes

$$\begin{aligned}\max_{\lambda} \quad & -\frac{1}{2}\mathbf{z}_l^{\top}\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}(\mathbf{C}_u - \mathbf{C}_u D(\lambda)\mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ & +\rho\lambda^{\top}\mathbf{e} \qquad\qquad (5.5) \\ \text{s. t.} \quad & \mathbf{C}_u^{-1} \succeq D(\lambda), \quad \lambda_i \geq 0, i = 1, \ldots, n_u\end{aligned}$$

The above formulation allows us to understand how the parameter ρ controls the strength of regularization provided by the unlabeled data. In the following, we will show that by adjusting the value of ρ, we derive a series of learning models.

5.1.1 Learning Model when $\rho = 0$

First, we consider the case of $\rho = 0$. We have the following theorem to indicate the relationship between the dual problem (5.5) and supervised SVM.

Theorem 2. *When $\rho = 0$, the optimization problem is reduced to a standard supervised SVM.*

Proof. It is not difficult to see that the optimal solution to (5.5) is $\lambda = \mathbf{0}$. As a result, $\omega(\mathbf{z}_l, \rho)$ becomes

$$\omega(\mathbf{z}_l, \rho = 0) \;=\; -\frac{1}{2}\mathbf{z}_l\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}C_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l$$

Substituting $\omega(\mathbf{z}_l, \rho)$ in (5.3) with the formulation above, the overall optimization problem becomes

$$\begin{aligned}\min_{\mathbf{z}_l, \delta} \quad & \frac{1}{2}\mathbf{z}_l^{\top}(\mathbf{C}_l^{-1} - \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}C_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1})\mathbf{z}_l + C\sum_{i=1}^{n_l}\delta_i \\ \text{s. t.} \quad & y_i z_i \geq 1 - \delta_i, \delta_i \geq 0, \quad 1 \leq i \leq n_l\end{aligned}$$

According to the matrix inverse lemma, we have $\mathbf{C}_l^{-1}$ calculated as

$$\begin{aligned}\mathbf{C}_l^{-1} &= (\mathbf{K}_{l,l} - \mathbf{K}_{l,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{u,l})^{-1} \\ &= \mathbf{K}_{l,l}^{-1} + \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}(\mathbf{K}_{u,u} - \mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u})^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1} \\ &= \mathbf{K}_{l,l}^{-1} + \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\end{aligned}$$

Hence, the final optimization problem is simplified as

$$\begin{aligned}\min_{\mathbf{z}_l,\delta} \quad & \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{z}_l + C\sum_{i=1}^{n_l}\delta_i \\ \text{s. t.} \quad & y_i z_i \geq 1 - \delta_i, \delta_i \geq 0, \quad 1 \leq i \leq n_l\end{aligned} \tag{5.6}$$

Clearly, the above optimization is identical to the standard supervised SVM. Hence, the unlabeled data is not used to regularize the decision boundary when $\rho = 0$. □

5.1.2 Learning Model when ρ is small

Second, we consider the case when ρ is small. According to (5.5), we expect λ to be small when ρ is small. As a result, we can approximate $(\mathbf{C}_u - \mathbf{C}_u D(\lambda)\mathbf{C}_u)^{-1}$ as follows

$$(\mathbf{C}_u - \mathbf{C}_u D(\lambda)\mathbf{C}_u)^{-1} \approx \mathbf{C}_u^{-1} + D(\lambda)$$

Consequently, we could write $\omega(\mathbf{z}_l, \rho)$ as follows:

$$\omega(\mathbf{z}_l, \rho) = -\frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathbf{C}_u^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l + \phi(\mathbf{z}_l, \rho) \tag{5.7}$$

where $\phi(\mathbf{z}_l, \rho)$ is the output of the following optimization problem

$$\begin{aligned}\max_{\lambda} \quad & \rho\lambda^\top \mathbf{e} - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}D(\lambda)\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ \text{s. t.} \quad & \mathbf{C}_u^{-1} \succeq D(\lambda), \quad \lambda_i \geq 0, i = 1, \ldots, n_u\end{aligned}$$

We can simplify the above problem by approximating $\mathbf{C}_u^{-1} \succeq D(\lambda)$ as $\lambda_i \leq [\sigma_1(\mathbf{C}_u)]^{-1}, i = 1, \ldots, n_u$, where $\sigma_1(\mathbf{C}_u)$ represents the maximum eigenvalue of matrix $\mathbf{C}_u$. The resulting simplified problem becomes

$$\begin{aligned} \max_{\lambda} \quad & \rho\lambda^\top \mathbf{e} - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}D(\lambda)\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ \text{s. t.} \quad & 0 \leq \lambda_i \leq [\sigma_1(\mathbf{C}_u)]^{-1}, 1 \leq i \leq n_u \end{aligned}$$

Note that the above problem is a linear programming problem, thus the solution for λ is

$$\lambda_i = \begin{cases} 0 & [\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l]_i^2 > \rho \\ \sigma(\mathbf{C}_u)^{-1} & [\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l]_i^2 \leq \rho \end{cases}$$

As indicated by the above formulation, ρ is used as the threshold to select the unlabeled examples. Since $[\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l]_i$ can be viewed as the approximate prediction for the ith unlabeled example, the above formulation can be interpreted in the way that only the unlabeled examples with low prediction confidence will be selected for regularizing the decision boundary. All the unlabeled examples with high prediction confidence will be ignored. Based on the above analysis, we see that the parameter ρ controls the strength of regularization by the unlabeled examples.

5.1.3 Learning Model when ρ is large

For the general case, the quantity $(\mathbf{C}_u - \mathbf{C}_u D(\lambda)\mathbf{C}_u)^{-1}$ can be expanded according to the matrix inverse lemma

$$(\mathbf{C}_u - \mathbf{C}_u D(\lambda)\mathbf{C}_u)^{-1} = \mathbf{C}_u^{-1} + (D(\lambda)^{-1} - \mathbf{C}_u)^{-1}$$

Thus, the function $\omega(\mathbf{z}_l, \rho)$ can be expanded as (5.7) and the $\phi(\mathbf{z})$ in (5.7) is the output of the following optimization problem

$$\begin{aligned} \max_{\lambda} \quad & \rho\lambda^\top \mathbf{e} - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}(D(\lambda)^{-1} - \mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ \text{s. t.} \quad & D(\lambda)^{-1} \succeq \mathbf{C}_u, \lambda \succeq 0 \end{aligned}$$

We can reparameterize the above problem by defining $\theta_i = 1/\lambda_i$ and rewrite the above problem as follows:

$$\max_{\theta} \quad \sum_{i=1}^{n_u} \frac{\rho}{\theta_i} - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}(D(\theta) - \mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \quad (5.8)$$

$$\text{s. t.} \quad D(\theta) \succeq \mathbf{C}_u, \lambda \succeq 0 \quad (5.9)$$

We can further simplify the above problem by approximating $D(\theta) \succeq \mathbf{C}_u$ as

$$\theta_i \geq \sum_{j=1}^{n_u} |[\mathbf{C}_u]_{i,j}|$$

The resulting formulation becomes

$$\max_{\theta} \quad \sum_{i=1}^{n_u} \frac{\rho}{\theta_i} - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}(D(\theta) - \mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \quad (5.10)$$

$$\text{s. t.} \quad \theta_i \geq \sum_{j=1}^{n_u} |[\mathbf{C}_u]_{i,j}|, 1 \leq i \leq n_u$$

When ρ is very large, we expect θ_i to be as small as possible, and therefore $\theta_i = \sum_{j=1}^{n_u} |[\mathbf{C}_u]_{i,j}|$. Thus, $D(\theta) - \mathbf{C}_u$ can be approximated by the combinatorial Laplacian of $\mathbf{C}_u$, i.e., $\mathcal{L}(\mathbf{C}_u)$. Finally, the overall optimization problem when ρ is large becomes:

$$\begin{aligned} \min_{\mathbf{z}_l, \delta} \quad & \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{z}_l - \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathcal{L}(\mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}\mathbf{z}_l \\ & + C\sum_{i=1}^{n_l} \delta_i \qquad (5.11) \\ \text{s. t.} \quad & y_i z_i \geq 1 - \delta_i, \ \delta_i \geq 0, \ 1 \leq i \leq n_l \end{aligned}$$

5.2 Understanding Manifold Regularization

The manifold regularization SVM [10] can be formulated as:

$$\begin{aligned} \min_{\mathbf{z}_l,\mathbf{z}_u,\delta} \quad & \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{z}_l + \frac{\gamma}{2}(\mathbf{z}_l^\top, \mathbf{z}_u^\top)\mathcal{L}(\mathbf{z}_l^\top, \mathbf{z}_u^\top) \\ & + C\sum_{i=1}^{n_l}\delta_i \qquad (5.12) \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l. \end{aligned}$$

If we regard the Laplacian $\mathcal{L}$ as the pseudo inverse of the kernel matrix $\mathbf{K}$, i.e., $\mathcal{L} = \mathbf{K}^*$ and replace $\mathbf{K}^*$ with $\mathbf{I}-\mathbf{K}$, we then have the following optimization problem.

$$\begin{aligned} \min_{\mathbf{z}_l,\mathbf{z}_u,\delta} \quad & \frac{1}{2}\mathbf{z}_l^\top \mathbf{K}_{l,l}^{-1}\mathbf{z}_l + \frac{\gamma}{2}(\mathbf{z}_l^\top, \mathbf{z}_u^\top)(\mathbf{I}-\mathbf{K})(\mathbf{z}_l;\mathbf{z}_u) + C\sum_{i=1}^{n_l}\delta_i \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l. \end{aligned}$$

In the above, we assume that the kernel similarity matrix $\mathbf{K}$ is normalized. The regularization term $(\mathbf{z}_l^\top, \mathbf{z}_u^\top)(\mathbf{I}-\mathbf{K})(\mathbf{z}_l;\mathbf{z}_u)$ is expanded as:

$$\begin{aligned} & (\mathbf{z}_l^\top, \mathbf{z}_u^\top)(\mathbf{I}-\mathbf{K})(\mathbf{z}_l;\mathbf{z}_u) \\ = \ & \mathbf{z}_l^\top(I-\mathbf{K}_{l,l})\mathbf{z}_l + \mathbf{z}_u^\top(I-\mathbf{K}_{u,u})\mathbf{z}_u - 2\mathbf{z}_u^\top\mathbf{K}_{u,l}\mathbf{z}_l \end{aligned}$$

By minimizing over $\mathbf{z}_u$, we have the above regularization term written as

$$\frac{1}{2}\mathbf{z}_l^\top(\mathbf{I}-\mathbf{K}_{l,l})\mathbf{z}_l - \frac{1}{2}\mathbf{z}_l\mathbf{K}_{l,u}(\mathbf{I}-\mathbf{K}_{u,u})^{-1}\mathbf{K}_{l,u}\mathbf{z}_l$$

Using the above expression, the overall formulation becomes

$$\begin{aligned} \min_{\mathbf{z}_l,\delta} \quad & \frac{1}{2}\mathbf{z}_l^\top\left(\mathbf{K}_{l,l}^{-1} + \gamma(\mathbf{I}-\mathbf{K}_{l,l})\right)\mathbf{z}_l + C\sum_{i=1}^{n_l}\delta_i \\ & -\frac{\gamma}{2}\mathbf{z}_l^\top\mathbf{K}_{l,u}(\mathbf{I}-\mathbf{K}_{u,u})^{-1}\mathbf{K}_{u,l}\mathbf{z}_l \qquad (5.13) \\ \text{s. t.} \quad & y_i z_i \geq 1-\delta_i,\ \delta_i \geq 0,\ 1 \leq i \leq n_l. \end{aligned}$$

Comparing the optimization problem in (5.13) with the optimization problem in (5.11), we see that these two problems are similar. The key difference is that $\mathbf{K}_{l,l}^{-1}$ in (5.11) becomes $\mathbf{K}_{l,l}^{-1} + \gamma(\mathbf{I} - \mathbf{K}_{l,l})$, and $\mathbf{K}_{l,l}^{-1}\mathbf{K}_{l,u}\mathcal{L}(\mathbf{C}_u)^{-1}\mathbf{K}_{u,l}\mathbf{K}_{l,l}^{-1}$ is replaced by $\mathbf{K}_{l,u}(\mathbf{I} - \mathbf{K}_{u,u})^{-1}\mathbf{K}_{l,u}$. Note that $\mathbf{C}_u \approx \mathbf{K}_{u,u}$ when the unlabeled examples are loosely related to the labeled examples (i.e., $\mathbf{K}_{u,l}$ are small in most of its elements). The key difference between TSVM and manifold regularization is that TSVM is able to select the unlabeled examples when constructing the regularizer using the unlabeled examples, while the regularizer used by the manifold regularization is independent from the labeled examples.

5.3 Summary

We consider the connection between two fundamental assumptions in semi-supervised learning. More specifically, we show that the loss on the unlabeled data employed by TSVM can essentially be viewed as an additional regularizer for the decision boundary. We further show that this additional regularizer induced by the TSVM is closely related to the regularizer introduced by the manifold regularization. Both of them can be viewed as a unified regularization framework.

□ **End of chapter.**

Chapter 6

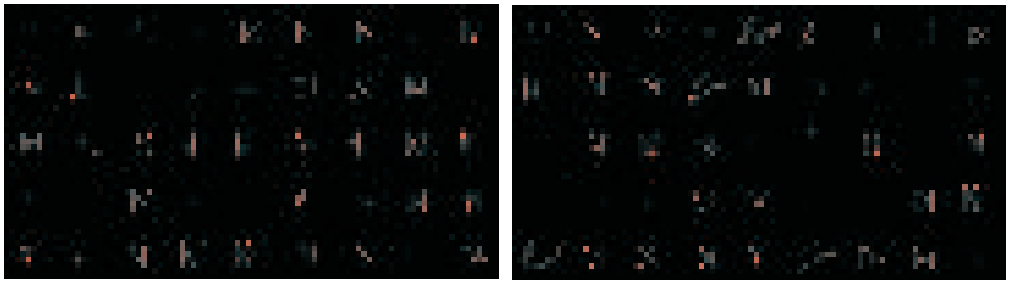
Learning from Weakly-related Unlabeled Data

In this chapter, we consider the case that the unlabeled data are in poor quality and are only structurally related to the labeled data in the current task. Following the first model, denoted as self-taught learning, dealing with such weakly-related data [112], We name the proposed framework as Supervised Self-taught Learning as we actively extract the high level features from the weakly-related unlabeled data with the supervision of the labeled of the learning task.

Self-taught Learning (STL) that is able to adequately utilize the information from the unlabeled data receives active attentions recently [112, 90]. Generally speaking, how to use the unlabeled data for learning is also the core topic for the so-called Semi-Supervised Learning (SSL) [163, 165, 159, 148]. However, SSL usually requires that the unlabeled data share the same data distribution with the labeled data. More specifically, the unlabeled data should contain the same labels as those of the labeled data within the SSL framework. Unfortunately, the unlabeled data which share the same distribution with the limited labeled data may sometimes be still difficult to obtain. Instead, a huge amount of seemingly irrelevant unlabeled data could be available at hand or in WWW in most cases. Self-taught Learning is proposed to deal with such difficult yet interesting problem.

To better understand the above background, we consider the automatic classification task of the images of dinosaurs and elephants. In this task, the labeled training samples are limited. Moreover, it is also quite expensive to obtain many unlabeled images of dinosaurs and elephants. Both the supervised learning and the SSL fail to solve this problem due to the limited amount of labeled and unlabeled dinosaur and elephant images. In contrast, STL is shown to be able to improve the classification performance by appropriately utilizing a huge amount of other unlabeled image samples, e.g., other types of animals or even natural scene pictures; these samples are seemingly irrelevant but can be easily obtained [112]. The motivation for STL is that, many randomly chosen images contain basic visual patterns, e.g., edges, which might be similar to those in images of dinosaurs and elephants. Another analogy is that handwritten digits can help to recognize the English characters, since the digits contain the strokes that are similar to those in the English characters, although they have the different distributions or labels. Studies in [112] have demonstrated that STL could be promising for the task mentioned above and can indeed improve the classification accuracy in some cases.

The learning procedure in STL can be divided into three separate stages. In the first stage, the high level representations (or basis), e.g., the edges in the images, or the strokes in the English characters, are learned from available unlabeled data which are unnecessarily relevant to the concepts of the labeled objects. In the second stage, STL represents the labeled data in a linear combination form of the high level features or basis obtained from the first stage. Those coefficients of the basis are then treated as the input features for the next stage. In the third stage, one can exploit traditional supervised learning algorithms, e.g., Support Vector Machine (SVM) [139], to learn a



(a) Patterns learned by STL

(b) Patterns learned by our proposed framework

Figure 6.1: High-level visual features extracted by STL and SSTL when classifying digits "1" and "7" with capital letters "I", "M", "N" as unlabeled samples. (a) and (b) presents the patterns learned by STL and SSTL respectively. In (a), STL fail to extract the discriminative features, i.e., horizontal stroke patterns. In (b), SSTL manages to learn many horizontal stroke patterns.

decision function based on the coefficients. These three stages are conducted step by step in an isolated style.

One major problem for the above Self-taught Learning framework is that its first stage is somewhat conducted in a hit-or-miss way. Concretely, the learned high-level features in this step is only determined by the unlabeled data; these data could be much different from the target samples, i.e., the labeled data. The leaned patterns might be unsuitable or even misleading for classifying the labeled data in the following two stages. To illustrate this shortcoming, we consider another typical example of classifying two digits "1" and "7". Suppose that we have a huge number of other unlabeled uppercase English characters "I", "M", and "N". Obviously, the vertical strokes dominates the other strokes and no explicit horizontal stokes occur in these three characters. Hence, the feature of the horizontal stroke may not even appear in the final high-level features learned from the unlabeled data. However, to classify "1" and "7", the most discriminate feature is the horizontal stroke. Figure 6.1(a) visually shows the 50 high-level features extracted by STL from 200 "I", "M", "N" characters. Clearly observed, almost no horizontal stroke patterns are extracted.

Aiming to solve the above problem, we propose a novel Supervised Self-taught Learning (SSTL) model which manages to find the most appropriate high-level features or representations from the unlabeled data under the *supervision* of the labeled training data. We attempt to learn from unlabel data with the "target" in mind rather than to achieve this in a hit-or-miss way. More specifically, the optimization is not separately performed as made in the traditional self-taught learning. Instead, three stages (the basis learning, coefficient optimization, and the classifier learning) are integrated into a single optimization problem. The representations, the coefficients, and the classifier are optimized simultaneously. By interacting the classifier optimization with choosing the high-level representations, the proposed model is able to select those discriminant features or representations, which are most appropriate for classification. Hence this will greatly benefit the classification performance. Figure 6.1(b) demonstrates the high-level basis obtained by our SSTL framework in the "1" and "7" classification problem. Evidently, the most discriminative patterns, the horizontal strokes, can indeed be extracted.

To our best knowledge, this is the first study that performs the Self-taught Learning in a supervised way. The underlying knowledge embedded in the unlabeled data can be transferred to the classification task actively and efficiently. In addition, one important feature of our novel framework is that the final optimization can be solved iteratively with the convergence guaranteed. Moreover, we show that the proposed discriminative framework can even be formulated into a single optimization problem for the multi-way classification tasks. With these two merits, the proposed framework can be easily applied in practice for many applications.

In the following, we first present the problem as well as the notations used throughout the chapter. We then review the Self-taught Learning algorithm in brief. After that, we present our novel Supervised Self-taught Learning framework. In Section 8.2, we provide a series of experiments to verify the proposed framework. In Section 6.5, we discuss some important issues. Finally, we set out the conclusion with some final remarks.

6.1 Problem Formalism

Given a labeled training data set $D = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \ldots, (\mathbf{x}^l, y^l)\}$, consisting of l labeled samples drawn i.i.d. from a certain distribution S. Here $\mathbf{x}^i \in \mathbb{R}^n$ $(i = 1, 2, \ldots, l)$ describes an input feature vector, and $y^i \in \{1, 2, \ldots, q\}$ is the category label for $\mathbf{x}^i$. In addition, assume that m $(m \gg l)$ unlabeled data samples $\{\mathbf{x}^{l+1}, \mathbf{x}^{l+2}, \ldots, \mathbf{x}^{l+m}\}$ are also available. The basic task of STL and SSTL can be informally described as seeking a hypothesis $h : \mathbb{R}^n \to \{1, 2, \ldots q\}$ that can predict the label $y \in \{1, 2, \ldots q\}$ for the future input data sample $\mathbf{z} \in \mathbb{R}^n$ by appropriately exploiting both the labeled data and those seemingly irrelevant unlabeled data.

Remarks. Note that the above problem is much different from the SSL. SSL requires that unlabeled data should be sampled from the same distribution of the labeled data; however, in the mentioned task, these unlabeled samples do not have such constraints. In other words, these unlabeled samples might share different labels from those of the labeled data. The problem is also much different from Transfer Learning (TL) [134, 42, 114] in that the latter framework requires these auxiliary data are already labeled.

6.2 Self-taught Learning by Sparse Coding

STL solves the above mentioned task in three separate stages. We describe these three stages in the following.

6.2.1 Stage I: Learning Representations

In the first stage, high-level representations are learned from the unlabeled data. For instance, edges could be learned from the natural scene images in the task of classifying dinosaur and elephants; strokes could be learned from the available English characters even if our purpose is to classify handwritten digits. These high-level representations can be learned by using Sparse Coding (SC). SC is a powerful technique that receives much interest recently. It can learn over-complete basis from data. We refer interesting readers to [117, 107, 91, 108]. The formulation is as follows:

$$\begin{aligned} \min_{\mathbf{a},\mathbf{b}} \quad & \sum_{i=l+1}^{l+m} \|\mathbf{x}^i - \sum_{j=1}^{p} a_j^{(i)} \mathbf{b}_j\|_2^2 + \beta \|\mathbf{a}^{(i)}\|, \\ \text{s. t.} \quad & \|\mathbf{b}_j\|_2^2 \leq 1, j = 1, \ldots, p \, . \end{aligned}$$

$\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p\}$ is called a set of *basis* with each basis $\mathbf{b}_j$ as an n-dimensional vector. $a_j^{(i)}$ is the activation coefficient associated with the basis $\mathbf{b}_j$ for the sample $\mathbf{x}^i$. Hence $\mathbf{a}^{(i)}$ ($i = l+1, l+2, \ldots, l+m$) is a set of activation coefficients for the unlabeled sample $\mathbf{x}^i$ with respect to all the basis $\mathbf{b}$. We denote $\mathbf{a}$ as a matrix defined as $(\mathbf{a}^{(l+1)}, \mathbf{a}^{(l+2)}, \ldots, \mathbf{a}^{(l+m)})$.

The above optimization problem tries to represent the unlabeled data in terms of $\mathbf{b}$. In more details, the first term in the optimization function describes the reconstruction error, while the second term using the L_1-norm forces the activation vector to be sparse. It is noted that the above optimization resembles the Principal Component Analysis (PCA) [47] if the second term is omitted. However, SC enjoys several advantages over PCA. First, PCA can only generate a limited number of basis (fewer than n), while SC could generate a large number of basis whose number might be far larger than n. Second, PCA only results in linear feature extraction, while SC can deliver non-linear representations as imposed by the L_1-norm. Containing such merits, SC is shown to be better than PCA in many cases and is actively adopted to learn over-complete representations from data [117, 107].

6.2.2 Stage II: Feature Construction from Basis

In the second stage, STL tries to represent the labeled data with respect to the basis $\mathbf{b}$. This stage is formulated as follows:

$$\min_{\mathbf{a}_L} \quad \sum_{i=1}^{l} \|\mathbf{x}^i - \sum_{j=1}^{p} a_{L_j}^{(i)} \mathbf{b}_j\|_2^2 + \beta \|\mathbf{a}_L^{(i)}\| \ .$$

In this stage, the features or the activation coefficients $\mathbf{a}_L$ for the labeled data are learned over the basis $\mathbf{b}$, which are obtained from the first stage. Similarly, the second term forces the coefficient vector a sparse form.

6.2.3 Stage III: Learning a Classifier from Features

In the third stage, an SVM can be exploited to learn the decision function $h = \mathbf{w} \cdot \mathbf{a_z} + c$ ($\mathbf{a_z}$ is the coefficient vector of the future sample $\mathbf{z}$) from the features constructed in Stage II. This is described in the following:

$$\begin{aligned} \min_{\mathbf{w},c} \quad & \sum_{i=1}^{l} \varepsilon_i + \gamma \|\mathbf{w}\|_2^2 \ , \\ \text{s. t.} \quad & y^k(\mathbf{w}.\mathbf{a}_L^{(k)} + c) \geq 1 - \varepsilon_k, \\ & \varepsilon_k \geq 0, k = 1, \ldots, l \ . \end{aligned}$$

Clearly, the above optimization problem is the standard L_2-norm Support Vector Machine, except that the input features are the coefficients obtained in the second stage. In real applications, L_1-norm SVM [162] can also be adopted.

Observed from the above optimization, STL extracts the high-level representations from the unlabeled data only. However, these high-level representations may be inappropriate or even misleading for the latter classifier construction. The discriminative information, that proves critical for classification performance, may be discarded in this stage. In the next section, we propose the Supervised Self-taught Learning framework that successfully integrates the above three stages into one optimization problem. In the new framework, the high-level representation optimization is supervised by the classifier learning. The derived representations would be those discriminative patterns that will greatly benefit the classification performance.

6.3 Supervised Self-taught Learning Framework

In this section, we present our novel Supervised Self-taught Learning framework. For the purpose of clarity, we first describe the framework in the binary setting. We then present how to extend the framework to multi-way classification.

6.3.1 Two-category Model

The binary SSTL model is formulated as the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{l+m} \|\mathbf{x}^i - \sum_{j=1}^{p} a_j^{(i)} \mathbf{b}_j\|_2^2 + \beta\|\mathbf{a}^{(i)}\| + \lambda \sum_{i=1}^{l} \varepsilon_i + \gamma\|\mathbf{w}\|_2^2 \ , \\ \text{s. t.} \quad & \|\mathbf{b}_j\|_2^2 \leq 1, j = 1, \ldots, p \ , \\ & y^k(\mathbf{w} \cdot \mathbf{a}^{(k)} + c) \geq 1 - \varepsilon_k, \varepsilon_k \geq 0, k = 1, \ldots, l \ . \end{aligned}$$

In the above, $\mathbf{b}_j, j = 1 \ldots, p$ represents p basis extracted from the unlabeled data under the supervision of the labeled data. $a_j^{(i)}$ is the weight or the coefficient for the data point $\mathbf{x}^i$ with respect to the basis $\mathbf{b}_j$. $\{\mathbf{w}, c\}$ defines the classifier boundary[1].

The optimization not only minimizes the reconstruction error among both the labeled data and unlabeled data given by $\sum_{i=1}^{l+m} \|\mathbf{x}^i - \sum_{j=1}^{p} \mathbf{a}_j^{(i)} \mathbf{b}_j\|_2^2$, but also minimizes the error $\sum_{i=1}^{l} \varepsilon_i$ caused by the classifier on the labeled data. An L_1-norm and an L_2-norm is respectively exploited as the regularization terms for $\mathbf{a}$ and $\mathbf{w}$, respectively. One can also use the L_1-norm for $\mathbf{w}$ in practice. The basis $\mathbf{b}$ (or the high-level features) and the classifier $\{\mathbf{w}, c\}$ are optimized simultaneously. Therefore, the extracted features will be more appropriate for classification. This is much different from the original self-taught framework using sparse coding, where the high-level features, determined exclusively by the unlabeled data, might be misleading and deteriorate the classification.

Similar to the original sparse coding problem, the above optimization problem is not convex. However, it is convex in $\mathbf{a}$ (while holding $\{\mathbf{b}, \mathbf{w}, c, \varepsilon\}$ fixed) and also convex in $\{\mathbf{b}, \mathbf{w}, c, \varepsilon\}$ (while holding $\mathbf{a}$ fixed). In the following, we show how to solve the optimization problem iteratively in two steps.

[1] For binary problems, we modify the class labels as {-1,+1}.

6.3.2 Optimization Method

We propose the following iterative algorithm to conduct optimization. When $\mathbf{b}, \mathbf{w}, c, \varepsilon$ are fixed, it is easy to verify that the optimization problem of finding $\mathbf{a}$ is reduced to the following two sub problems.

Problem I(a):

$$\min_{\mathbf{a}^{(i)}} \quad \|\mathbf{x}^i - \sum_{j=1}^{p} \mathbf{a}_j^{(i)} \mathbf{b}_j\|_2^2 + \beta \|\mathbf{a}^{(i)}\|, \; i = l+1, \ldots, l+m \; .$$

Problem I(b):

$$\begin{aligned} \min_{\mathbf{a}^{(i)}} \quad & \|\mathbf{x}^i - \sum_{j=1}^{p} \mathbf{a}_j^{(i)} \mathbf{b}_j\|_2^2 + \beta \|\mathbf{a}^{(i)}\|, i = 1, \ldots, l, \\ \text{s. t.} \quad & y^i(\mathbf{w} \cdot \mathbf{a}^{(i)} + c) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0 \; . \end{aligned}$$

Problem I(a) describes the optimization over unlabeled data, while I(b) presents the optimization over the labeled data points. Problem I(a) is equivalent to a regularized least squares problem; I(b) is similar except that it has a linear constraint. Both problems can be practically solved by many algorithms, e.g., the feature-sign search algorithm [91], the interior point method [105], or a generic convex programming solver (CVX)[2].

Similarly, when $\mathbf{a}$ is fixed, the optimization problem of finding $\mathbf{b}, \mathbf{w}, c, \varepsilon$ is changed to the following two sub problems.

Problem II(a):

$$\begin{aligned} \min_{\mathbf{w}, c, \varepsilon} \quad & \gamma \|\mathbf{w}\|_2^2 + \lambda \sum_{k=1}^{l} \varepsilon_k, \\ \text{s. t.} \quad & y^k(\mathbf{w} \cdot \mathbf{a}^{(k)} + c) \geq 1 - \varepsilon_k, \varepsilon_k \geq 0, k = 1, \ldots, l \; . \end{aligned}$$

[2] The matlab source codes of the CVX package can be downloaded from http://www.stanford.edu/ boyd/cvx/.

Problem II(b):

$$\begin{aligned} \min_{\mathbf{b}} \quad & \sum_{i=1}^{l+m} \|\mathbf{x}^i - \sum_{j=1}^{p} a_j^{(i)} \mathbf{b}_j\|_2^2 \ , \\ \text{s. t.} \quad & \|\mathbf{b}_j\|_2^2 \leq 1, j = 1, 2, \ldots p \ . \end{aligned}$$

Problem II(a) and II(b) are typical quadratic programming problems. More specifically, II(a) is the standard L_2-norm SVM optimization problem; II(b) is a Quadratic Constrained Quadratic Programming problem (QCQP) [8, 98, 13]. They can be either solved by the SMO algorithm [111] or the dual algorithm proposed in [91].

Since the value of the optimization objective $f(\mathbf{a}, \mathbf{b}, \mathbf{w}, \varepsilon)$ will be decreased after solving each problem, solving the above two problems alternatively will guarantee a convergence to a fixed point. As a summary, we present the optimization algorithm in Algorithm 3.

6.3.3 Multi-category Model

In this section, we provide the details of how to exploit the one-against-others strategy to extend our SSTL to multi-way tasks.

Before we present the problem definition for the multi-category model, we define some notations in the following. Let $\mathbf{I}_o$ be a diagonal matrix with the element in (o, o) as 1 and all the other diagonal elements as -1. Assume that $\{\mathbf{w}_o, c_o\}$ be the decision function associated with the o-th class ($1 \leq o \leq q$, i.e., there are q categories) . We further define $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q)$ and $\mathbf{c} = (c_0, \ldots, c_q)^T$.

The multi-category SSTL model is defined as follows:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{l+m} \{ \|\mathbf{x}^i - \sum_{j=1}^{p} \mathbf{a}_j^{(i)} \mathbf{b}_j\|_2^2 + \beta \|\mathbf{a}^{(i)}\| \} + \lambda \sum_{i=1}^{l} \sum_{o=1}^{q} \varepsilon_o^i \\
& + \gamma \sum_{o=1}^{q} \|\mathbf{w}_o\|_2^2 \qquad (6.1) \\
\text{s. t.} \quad & \|\mathbf{b}_j\|^2 \leq 1, j = 1, \ldots, p \ , \\
& \mathbf{I}_o(\mathbf{W}^T \mathbf{a}^{(k)} + \mathbf{c}) \geq \mathbf{e} - \varepsilon^k, \\
& \varepsilon^k \geq 0, k = 1, \ldots, l \ .
\end{aligned}$$

In the above, ε^k represents a q-dimensional slack vector for the k-th labeled data sample. Each element of ε^k, i.e., ε_o^k $(1 \leq o \leq q)$ represents the hinge loss incurred by the classifier $\{\mathbf{w}_o, c_o\}$ with respect to $\mathbf{x}^k$. $\varepsilon^k \geq 0$ means each element of ε^k is not less than 0. $\mathbf{e}$ is a vector with all the elements as one. Other variables are similarly defined as those in the binary case.

We now interpret the above multi-category model in the following. First, in binary classification, each labeled sample is used only once. However, in multi-way classification, each labeled sample will be used by q times, since there are q classifiers. Hence the hinge loss for each labeled sample is not a scale variable anymore. Instead, it is a q-dimensional vector. Second, the key point of multi-way classification training using Sparse Coding is to derive a common set of basis for all the q classifiers involved. This requires that the single optimization be formulated for q classifiers. Our model successfully achieves this goal. Finally, as observed from the above model, the optimization can still be optimized in two steps. Moreover, each step is easily verified to be convex as well. Hence it can be solved using the similar method as presented in the previous section.

Algorithm 3 Supervised Self-taught Learning Via Sparse Coding

```
Input: Labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$; unlabeled data $\{(\mathbf{x}_i, y_i)\}_{i=l+1}^{l+m}$
Step 1. Initialize $\mathbf{a}_{(0)}$; set $\Delta$ to a small positive value; set the number of iterations $t = 1$.
Step 2. Compute $\{\mathbf{w}_{(t)}, c_{(t)}, \varepsilon_{(t)}, \mathbf{b}\}$.
    a. Calculate $\{\mathbf{w}_{(t)}, c_{(t)}, \varepsilon_{(t)}\}$ by solving Problem II(a).
    b. Calculate $\mathbf{b}_{(t)}$ by solving Problem II(b).
Step 3. Compute $\{\mathbf{a}_{(t)}^{(i)}\}_{i=1}^{l+m}$.
    a. Calculate $\{\mathbf{a}_{(t)}^{(i)}\}_{i=l+1}^{l+m}$ by solving Problem I(a).
    b. Calculate $\{\mathbf{a}_{(t)}^{(i)}\}_{i=1}^{l}$ by solving Problem I(b).
Step 4. If $t < T_{\mathrm{MAX}}$ and $\|f_{(t)}(\mathbf{a}, \mathbf{b}, \mathbf{w}, \varepsilon) - f_{(t-1)}(\mathbf{a}, \mathbf{b}, \mathbf{w}, \varepsilon)\| > \Delta$, then $t \leftarrow t + 1$; go to Step 2; otherwise stop.
Output: The classifier $\{\mathbf{w}, c\} \leftarrow \{\mathbf{w}_{(t)}, c_{(t)}\}$
```

6.4 Experiments

In this section, we evaluate our proposed Supervised Self-taught Learning algorithm on various data. We first present a toy example in order to illustrate the proposed model clearly. We then report the experimental results on character images and web data.

6.4.1 Toy Example

We generate a toy example to demonstrate the merits of our proposed SSTL framework. The task is designed to discriminate handwritten numerals “1” and “7”. Suppose we have only 4 numerals which are drawn in the top area of Figure 6.2. Furthermore, we have an unlabeled data set which consists of 200 printed capital characters “I”, “M”, “N” . 20 examples for the

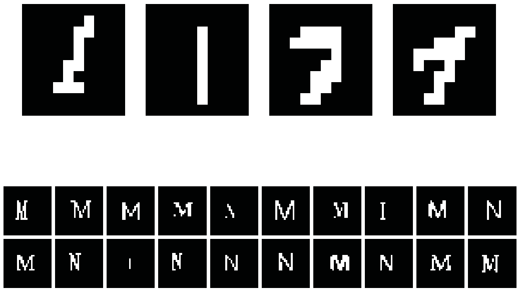


Figure 6.2: Examples of labeled data and unlabeled data used in the toy example. The top sub-figure contains 4 numerals, labeled as “1” for the first two characters, and “7” for the last two numerals. The bottom sub-figure provides 20 examples randomly extracted from 200 28×28 unlabeled English characters representing “I”, “M”, and “N”. The basis number is set to 50. λ, γ, β, are all set to 10.

unlabeled data are shown in the bottom of Figure 6.2. The test data set contains 500 numerals “1” and “7” which are randomly extracted from the MNIST data set [3]. We perform traditional supervised learning, i.e., the L_2-norm SVM, Self-taught Learning and Supervised Self-taught Learning on this data set. The used features are raw pixel intensity values. The results are shown in Table 6.1.

Table 6.1: Experimental results on classifying “1” and “7” with unlabeled characters “I”, “M”, “N”

“I”,“M”,“N” → “1” and “7”			
Method	SVM	STL	SSTL
Accuracy	83.07	78.23	**85.09**

From the results, we have several observations. First, although there are only 4 labeled samples for “1” and “7”, the supervised algorithm, SVM, still achieves the accuracy of 83.07% partly because “1” and “7” are originally easy to be discriminated. Second, the Self-taught Learning algorithm deteriorates

[3] http://yann.lecun.com/exdb/mnist/

the accuracy after it transfers the knowledge from the unlabeled data. As analyzed before, the most prominent features to separate "1" and "7" are the horizontal strokes; however, no explicit horizontal stroke patterns occur in the unlabeled data. This makes the patterns learned from STL lose some important discriminative information. This phenomenon can be observed if one looks back into Figure 6.1. Finally, our proposed Supervised Self-taught Learning algorithm can deal with this problem appropriately. It demonstrates the best performance clearly. It improves the purely supervised learning approach over 2 percent.

6.4.2 Handwritten numerals → English Characters

We also examine the performance of our proposed approach on the English character recognition task [4]. The unlabeled data are handwritten numerals of MNIST. We evaluate whether the unlabeled data can help to improve the classification performance when the number of labeled data is set to 100, 500, and 1000 respectively. The basis number is chosen via cross validation from $\{50, 100, 150, 200\}$. The parameter β is chosen from {10, 20, 50, 100}. γ and λ are searched in the range $\{5, 10, 50, 100\}$. For computational reasons, the unlabeled data are preprocessed by applying PCA to reduce their dimensionality. We follow [113] and set the number of principal components to keep approximately 96% of the unlabeled data variance. The final results are the average over 5 runs.

We report the results in Table 6.2. Once again, we observe that the proposed SSTL outperforms the STL algorithm and the supervised classifier trained only on the labeled data. In more details, the Self-taught Learning algorithms including STL and SSTL deliver better performance than the supervised algo-

[4] http://ai.stanford.edu/~btaskar/ocr/

Table 6.2: Experimental results on classification of 26 English characters with unlabeled handwritten numerals data

Handwritten numerals → English Characters			
Training Size	SVM	STL	SSTL
100	39.57	39.98	**41.43**
500	54.98	56.27	**58.72**
1000	61.26	63.49	**65.76**

rithm, SVM. This shows that using unlabeled data in a self-taught way can boost the performance of traditional supervised learning algorithms. Furthermore, SSTL focuses on using the transferred knowledge from unlabeled data in a discriminative way; it demonstrates further improvements over STL.

6.4.3 Web Text Categorization

In this section, we evaluate the SSTL framework on web text categorization tasks. We adopt four subsets of text documents for the evaluation from three benchmark text collections, namely WebKB [5], Reuters-21578 [6], and Ohsumed [7]. In the selected data sets, *course* vs. *non-course*, which is obtained from the WebKB corpus, contains course web pages and non-course web pages of several universities. The *bacterial* vs. *virus* data and the *male* vs. *female* data are extracted from the Ohsumed database that is a set of references from MEDLINE, the on-line medical information database, consisting of titles and/or abstracts from medical journals. The *grain* vs. *wheat* data set is from the Reuters-21578 Text Categorization Collection, which is a collection of documents that appeared on Reuters newswire in 1987. The description of the four selected data sets can be found in Table 8.1.

[5] http://www.cs.cmu.edu/∼webkb/

[6] http://www.daviddlewis.com/resources/testcollections/

[7] ftp://medir.ohsu.edu/pub/ohsumed

Table 6.3: Descriptions for the web text documents data

Corpus	Labeled Data	# Documents
WebKB	*course* vs. *non-course*	1051
Ohsumed	*bacterial* vs. *virus*	581
	male vs. *female*	871
Reuters	*grain* vs. *wheat*	865

We conduct two sets of experiments. In the first set of experiments, we randomly select 4 labeled documents from each data set to form the training set, and use the remaining documents as the test set. In the second set of experiments, 10 labeled documents are randomly selected to form the training set. In order to generate the unlabeled data for self-taught learning algorithms, we first select the keywords from the given training data and then retrieve the Internet to get a set of unlabeled web pages using the keywords as the query terms. Here Google is used as the search engine and we select top 1000 returned web pages as the unlabeled data for each data set. We then represent each document by a vector of term frequency. We select 500 most informative features according to their correlation to the text categories. Note that, due to both the inaccuracy of query keywords and the ambiguity of the searching engines, the returned web pages contain many irrelevant documents. SSL cannot be directly applied in this task. The parameters are similarly selected as mentioned in the previous subsection. And the final results are the average over 10 runs using the above training and testing process.

The experimental results are listed in Table 6.4. As observed from the results, STL indeed increases the recognition accuracies of supervised learning in some data sets, e.g., *male* vs. *female* when the training size is equal to 4. However, in many cases, STL demonstrates much worse performance than the supervised learning, e.g., in *course* vs. *non-course* and *grain* vs. *wheat* when the training size is equal to 4. Because web documents are usually of both high-dimension and of high sparsity, without supervision from the labeled data, it is very possible that STL extracts non-important or even noisy basis from the unlabeled data. This explains why STL sometimes deteriorates the performance. In comparison, our proposed SSTL successfully avoids this problem. SSTL attempts to detect those most discriminative patterns as the basis by supervising the self-taught learning process via the labeled data. As clearly seen in Table 6.4, SSTL is consistently better than or as the same as STL and SVM in all the four data sets. The difference between SSTL and the other two algorithms is more distinct in the *course* vs. *non-course* data set: the accuracy of SSTL is almost as twice as that of SVM, and is also significantly higher than that of STL. The experimental results clearly demonstrate the advantages of our proposed new learning framework.

Table 6.4: Comparisons on web text categorization tasks. STL performs worse than SVM usually due to the inappropriate high-level representations learned. SSTL presents the best results consistently by incorporating the knowledge selectively and discriminatively.

	Training Size= 4			**Training Size**= 10		
Data Set	SVM	STL	SSTL	SVM	STL	SSTL
course vs. *non-course*	39.48	34.39	**78.19**	45.18	87.48	**91.21**
bacterial vs. *virus*	61.82	53.42	**62.49**	**73.14**	72.79	**73.14**
male vs. *female*	52.49	64.70	**65.5**2	63.41	53.66	**68.25**
grain vs. *wheat*	57.63	51.93	**61.52**	65.39	67.02	**69.38**

6.5 Discussion

We discuss some important issues in this section. First, the Self-taught Learning framework is much different from many other current learning paradigms. The core idea of STL is how to boost the classification performance when the labeled data is limited by appropriately transferring the knowledge from those seemingly irrelevant unlabeled data. This is much different from the Semi-supervised Learning algorithms in that SSL requires the unlabeled data follow the same distribution as the labeled data; it is also different from the Transfer Leaning algorithms in that TL can only transfer knowledge from labeled data. Our proposed Supervised Self-taught Learning algorithm is still located in the self-taught learning paradigm, but it focuses on transferring the knowledge from unlabeled data in a supervised or discriminative way. In other words, SSTL proposes to extract "useful" information from unlabeled data that aims to improve the classification performance. This is much different from the traditional Self-taught Learning algorithm in the sense that STL transfers knowledge from unlabeled data in a unsupervised or even driftless way.

Second, it is not new to combine discriminative learning algorithms into the so-called generative or unsupervised learning framework [70, 72, 62, 55, 65]. Our proposed SSTL is also motivated from this idea. However, these methods are still supervised learning algorithms because they perform such hybrid learning only for the labeled data. In contrast, our proposed algorithm tries to learn discriminative information from unlabeled data. This is the major difference between our algorithm and these hybrid methods. In addition, we believe the hybrid techniques specially designed for supervised learning could also be applied in the SSTL framework. More specifically, we notice that the discriminative sparse coding algorithm proposed in [55] might be used in order to further improve the classification accuracy. We leave this topic as future work.

Third, we only focus on studying the Supervised Self-taught Learning framework by applying the sparse coding algorithm. Obviously, there are a lot of other algorithms that could be applied to this new learning framework. It is interesting to investigate how other existing algorithms can be adapted to the SSTL framework.

Finally, although we have successfully integrated the three isolated optimization problems of STL into a single optimization task, it introduces several extra parameters in order to balance the reconstruction errors in the unlabeled data and the optimization values contributed by the classifier learning. Currently, these parameters are tuned manually or by cross validation. It is desirable that some more efficient algorithm can be developed so as to speed up the parameter selection process. We leave this task as an open problem.

6.6 Summary

In this chapter, we have presented a study on the Supervised Self-taught Learning framework, which can transfer knowledge from unlabeled data actively. This framework successfully integrates the three-step optimization into a single optimization problem. By interacting the classifier optimization with choosing the high-level representations, the proposed model is able to select those discriminant features or representations, which are more appropriate for classification. Hence this may benefit the classification performance greatly. To our best knowledge, this is the first work that performs Self-taught Learning in a supervised way. We have demonstrated that the novel framework boils down to solving four sub optimization problems iteratively, each of them being convex. Moreover, the final optimization can be iteratively solved with the convergence guaranteed. Extensive evaluations on various data sets including character image data and web text data have shown that our proposed algorithm can improve the classification performance against the traditional Self-taught Learning algorithm and the supervised learning algorithm when the number of the labeled data is limited.

□ **End of chapter.**

Chapter 7

Learning from a Mixture of Unlabeled Data

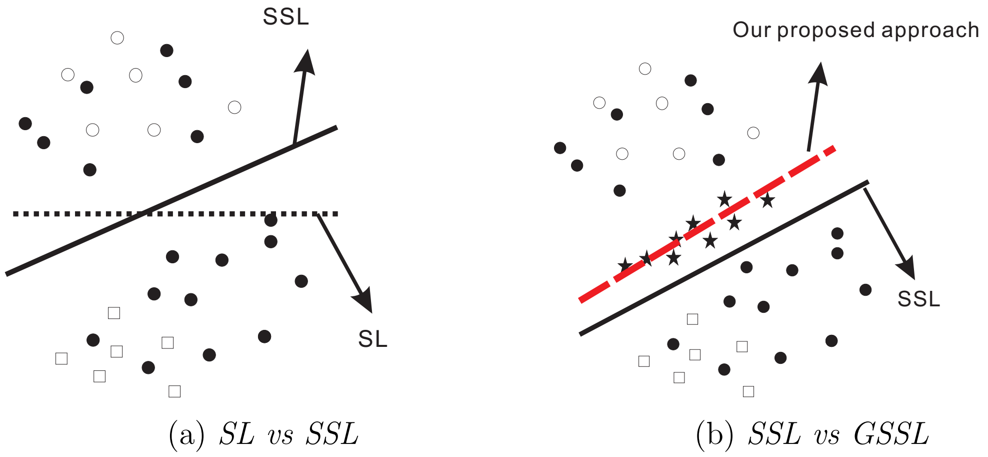
Learning classifiers from data has been a popular and important topic in machine learning and data mining. Given a sufficiently large quantity of labeled instances called training data, one can exploit the traditional Supervised Learning (SL) algorithms to handle this task [139, 48, 66]. However, in many real world applications, the labeled data may be very few due to the expensive cost of manual labeling. On the other hand, the number of unlabeled instances could be very large since they are generally much easier to obtain. Supervised learning, taking only advantages of the labeled data, might not work appropriately in these cases. In contrast, Semi-supervised Learning (SSL), making use of both labeled data and unlabeled data, proves to be an effective solution in addressing this problem [164, 28]. Undoubtedly, semi-supervised learning has achieved a great success in many domains involving machine learning and data mining. To guarantee good performance, semi-supervised learning usually assumes that the unlabeled data should share the same labels as the labeled training samples. Although this assumption can be well satisfied in some cases, it appears still strong in certain other domains. In fact, it is very common that unlabeled data are collected by using automatical tools. This is actually frequently seen in the earlier stages of data collection. It is usually inevitable that those collected unlabeled data contain irrelevant samples. Feeding such "corrupted" unlabeled data to semi-supervised learning may significantly affect the overall performance and incur severe problems consequently.

To attack this problem, we aim to build up a general semi-supervised learning framework capable of learning from general unlabeled data systematically, where the unlabeled data may contain irrelevant samples. Our model manages to better utilize the information from unlabeled data by formulating them as a three-class $(-1, +1, 0)$ mixture.[1] This hence distinguishes our work from the traditional semi-supervised learning problem where unlabeled data are assumed to contain the same labels as the labeled training samples [163, 46].

The benefits of taking the irrelevant data into account can be seen in Figure 1 and Figure 2. In both Figures, all the filled points ($\bullet$'s and $\star$'s) are unlabeled data, while the $\circ$'s and $\Box$'s are the two classes of labeled training samples. Clearly, Figure 1(a) illustrates that SSL can outperform the boundary given by the Support Vector Machines (SVM) [24, 139], the current state-of-the-art SL algorithm. However, SSL may encounter problems if the unlabeled data contain the "irrelevant" data. This can be observed in Figure 1(b): The boundary of SSL is obviously unreasonable. A more reasonable decision plane should pull away the "relevant" data (maximizing the margin among the negative and positive data) while predicting the values of the "irrelevant" data as close to zero as possible (clustering the "0"-data around the decision line). Such a boundary (the dashed red line) can be observed in Figure 1(b).

Exploiting the unlabeled data neither positive nor negative can actually remedy the negative impact when both the unla-

[1] In this chapter, we only consider the binary cases while multi-way problems can be easily approached via standard techniques, e.g., the one vs others technique [60].

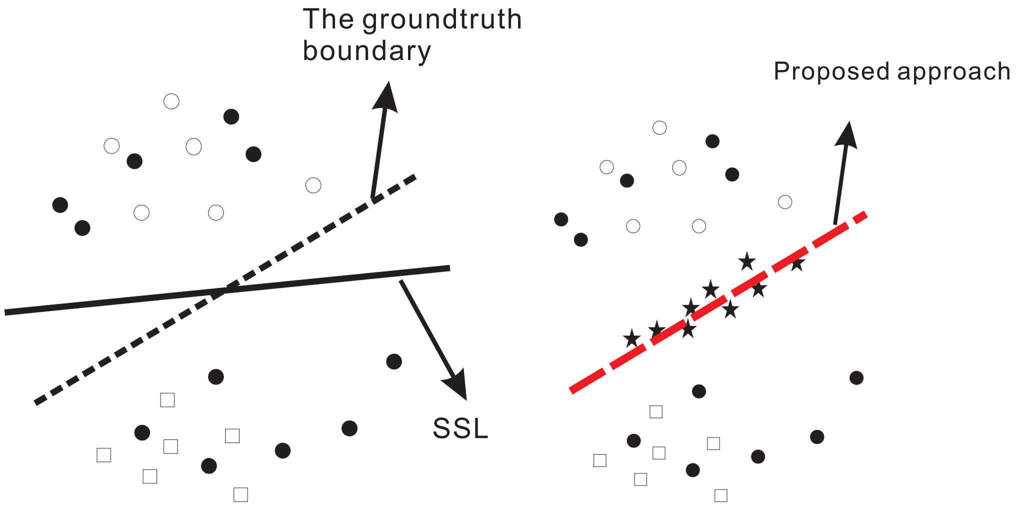


(a) *SL vs SSL* (b) *SSL vs GSSL*

Figure 7.1: The "irrelevant" data $\star$'s can increase the performance of the SSL. The filled points ($\bullet$'s and $\star$'s) are unlabeled data, while the $\circ$'s and $\Box$'s are the two classes of labeled training samples. The filled $\star$'s describe the irrelevant unlabeled data. The decision planes of the SL and SSL are given by the SVMs.

beled data and the labeled data are limited. Such a case can be seen in Figure 2. Assume the ground truth boundary is given as the dashed line in Figure 2(a). However, due to the limited training data (including both the labeled and relevant unlabeled data), the learned SSL boundary may be deviated from the actual one (as observed in Figure 2(a)). Sometimes, there are perhaps some "irrelevant" instances ($\star$'s in Figure 2(b)), being neither positive nor negative, mixed into the unlabeled data. By appropriately detecting and using these irrelevant data (trying to cluster such irrelevant unlabeled data around the decision plane), one can actually learn a more reasonable boundary as seen in Figure 2(b).

The idea of learning with the irrelevant data is similar to the work proposed in [126, 140], where the irrelevant data are called *universum*. However, they designed their system only within the Supervised Learning framework. In addition, these universum data need to be specified beforehand and are merely used as the labeled third class of samples. In other words, one needs to know which instances are universum data in advance so as to build a decision boundary. In comparison, we propose to exploit such irrelevant data in the semi-supervised context. More importantly, we do not need to specify which samples belong to the universum. Instead, we can learn from general unlabeled data, which means those relevant data or irrelevant data are mixed in the unlabeled data. Our novel model can output a more reasonable decision boundary, while simultaneously detecting the relevant data and irrelevant data automatically after the learning is finished.



(a) *SSL* (b) *GSSL*

Figure 7.2: The "irrelevant" data ⋆'s can increase the performance when only a limited number of relevant unlabeled data is available. The filled points (•'s and ⋆'s) are unlabeled data, while the ∘'s and □'s are the two classes of labeled training samples. The filled ⋆'s describe the irrelevant unlabeled data. The decision plane of the SSL is given by the SVM.

Indeed, as far as we know, this work presents a novel study on how to perform learning from general unlabeled data consisting of both relevant and irrelevant instances. When the irrelevant data are known as prior knowledge by the user, this is the idea of "SSL with universum" proposed in [157]. In contrast, our work presents a more difficult and general SSL framework, where irrelevant data are mixed with the relevant unlabeled data, without any knowledge on which samples are relevant or irrelevant beforehand. As a major contribution, we successfully formulate such a difficult problem as a Semi-definite Programming (SDP) problem [86, 45, 133], making the framework solvable in polynomial time. Both theoretical analysis and empirical investigations demonstrate that the proposed framework outperforms the traditional semi-supervised learning in many cases.

We detail the proposed framework including the model definition, the theoretical analysis, and the practical solving method in Section 7.1. In this section, we will demonstrate how the proposed model can be formulated in a Mixed Integer Programming (MIP) problem [99] and finally relaxed to be an SDP problem. In Section 7.2, we conduct a series of experiments to validate our novel approach. Finally, we set out the conclusion with final remarks.

7.1 Model

In this section, we first present the problem definition and the notation used in the chapter. We then introduce the model definition, the theoretical analysis and the practical solving method in turn.

7.1.1 Problem Formalism

Given a training data set $D \in \mathcal{R}^{l\times n}$, denoted by

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\},$$

drawn i.i.d. from a certain distribution S. Here $\mathbf{x}_i \in \mathcal{R}^n$ $(i = 1, 2, \ldots, l)$ describes an input feature vector, and $y_i \in \{-1, +1\}$ is the category label for $\mathbf{x}_i$. In addition, assume that m $(m \gg l)$ unlabeled data samples $\{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_{l+m}\}$ are also available (for brevity, we denote $n = l + m$). The unlabeled data contain both the relevant data sharing the same labels i.e., $\{-1, +1\}$ as the labeled data, and the irrelevant data which are different from the labeled data. Moreover, there are no prior knowledge on which instances are relevant or irrelevant.

The basic task here can be informally described as seeking a hypothesis $h : \mathcal{R}^n \rightarrow \{-1, +1\}$ that can predict the label $y \in \{-1, +1\}$ for the future input data sample $\mathbf{z} \in \mathcal{R}^n$ sampled from S by appropriately exploiting both the labeled data and the general unlabeled data. The hypothesis usually takes the linear form of $h = sign(f(\mathbf{z}))$, where $f(\mathbf{z}) = \mathbf{w}^\top \mathbf{z} + b$ ($\mathbf{w} \in \mathcal{R}^n$, $b \in \mathcal{R}$). Note that the linear form can be easily extended to the non-linear form based on the standard kernelization trick [119].

7.1.2 Framework

The novel framework is introduced in the following. We first present the model definition followed by the theoretical analysis showing the inner justifications of our model. Finally, we show how to transform the problem to an SDP problem.

Model Definition

The novel model is formulated as the following Problem I:

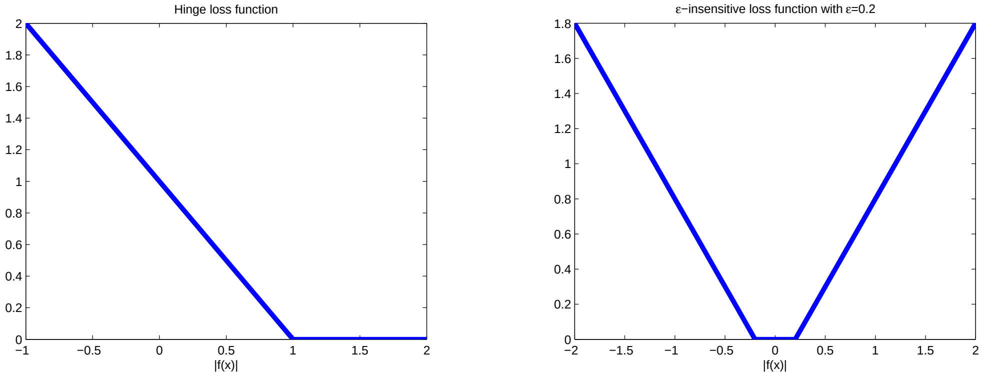
Problem I:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,\eta,\mathbf{y}_{l+1:n}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C_L \sum_{i=1}^{l} \xi_i + C_U \sum_{j=l+1}^{n} \min(\eta_j, \xi_j) \\
\text{s.t.} \quad & y_i(\mathbf{w}_i \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \ldots, l, & (7.1) \\
& y_j(\mathbf{w}_j \cdot \mathbf{x}_j + b) \geq 1 - \xi_j, & (7.2) \\
& |\mathbf{w}_j \cdot \mathbf{x}_j + b| \leq \varepsilon + \eta_j, & (7.3) \\
& \eta_j \geq 0, j = l + 1, \ldots, n, \\
& \xi_k \geq 0, k = 1, \ldots, n,
\end{aligned}$$

where $\mathbf{x}_i$, $i = 1, \ldots, l$ are the labeled training samples. Namely, $y_i \in \{-1, +1\}$ $i = 1, \ldots, l$ is known beforehand. $\mathbf{x}_j$, $j = l + 1, \ldots, n$ are the unlabeled data, where the associated labels are unknown, but restricted in the set of $\{-1, 0, +1\}$. C_L and C_U are two positive penalty parameters used to trade-off the margin and the training loss. ε is a small positive parameter describing the insensitiveness level.

Constraint (7.1) describes the loss for the labeled data. Constraint (7.2) provides the loss if $\mathbf{x}_j$ is judged as the ± 1 (i.e., the relevant data), while (7.3) presents the loss if $\mathbf{x}_j$ is judged as the class of 0 (i.e., the irrelevant class). The loss incurred by the unlabeled sample $\mathbf{x}_j$ is finally given by the minimum loss that it is judged as the class of ± 1 or 0. This can be seen in the objective function of Problem I. Intuitively, the above model attempts to maximize the margin among the positive relevant data and negative relevant data, while predicting the values of the irrelevant data as close to zero as possible simultaneously. In addition, our model can automatically detect or assign the unlabeled samples to either ± 1 (relevant classes) or 0 (irrelevant class) by choosing the smaller cost associated with the assigned label.

Note that two types of loss functions are adopted in Problem I. The loss function for the relevant data is the hinge loss $H_{-\varepsilon} = \max\{0, t - \varepsilon\}$ as seen in (7.2), where $t = 1$. On the other hand, the loss function of the irrelevant data is defined as the ε-insensitive loss $U[t] = H_{-\varepsilon}[t] + H_{\varepsilon}[t]$. Both loss functions are plotted in Figure 7.3. When a data point is judged as a relevant instance, we should push it as faraway as possible from the margin $f(\mathbf{z}) = \pm 1$. Hence a hinge loss is more appropriate for such a setting. When the data point belongs to the irrelevant class, it should be around the decision plane $f(\mathbf{z}) = 0$. In this sense, an ε-insensitive loss function is more suitable. An analogy can also be seen in choosing the loss functions for SVM (using hinge loss) and Support Vector Regression (using ε-insensitive loss) [128].



(a) *Hinge loss* (b) ε-*insensitive loss*

Figure 7.3: Hinge loss and ε-insensitive loss

It is not easy to directly optimize Problem I because of the operator of min. However, by introducing an integer variables $d_j = \begin{cases} 0 & \text{if} \quad y_j = \pm 1 \\ 1 & \text{if} \quad y_j = 0 \end{cases}$, $\forall j, l+1 \leq j \leq n$, we can transform Problem I to the following problem:

Problem II:

$$\begin{array}{rll} \min\limits_{\mathbf{w},b,\xi,\eta,\mathbf{y}_{l+1:n},\mathbf{d}} & \frac{1}{2}||\mathbf{w}||^2 + C_L \sum\limits_{i=1}^{l} \xi_i + C_U \sum\limits_{j=l+1}^{n} (\eta_j + \xi_j), & (7.4) \\ \text{s.t.} & y_i(\mathbf{w}_i \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \ldots, l & (7.5) \\ & y_j(\mathbf{w}_j \cdot \mathbf{x}_j + b) + \xi_j + M(1 - d_j) \geq 1, & (7.6) \\ & |\mathbf{w}_j \cdot \mathbf{x}_j + b| \leq \varepsilon + \eta_j + M d_j, & (7.7) \\ & d_j = \{0, 1\} \quad j = l+1, \ldots, n, & \\ & \eta_j \geq 0, j = l+1, \ldots, n, & \\ & \xi_k \geq 0, k = 1, \ldots, n. & \end{array}$$

In the above, M is a large positive constant. When d_j is equal to 0, $M(1 - d_j) = M$ is a big value. Hence (7.6) will naturally be satisfied, leading $\xi_i = 0$ and further $\min(\xi_j, \eta_j) = \xi_j + \eta_j$. A similar analysis can be obtained when $d_j = 1$. Therefore, we can know that Problem II is strictly equivalent to Problem I, provided that M is set to a sufficiently large value. Problem II is a Mixed Integer Programming problem [12, 99].

In the literature, there are a lot of proposals which can solve the MIP problem. In the following, we will first derive a theorem showing the justification of our proposed algorithm. We then revisit the optimization and propose our practical solving method.

Analysis

In this section, we conduct some analysis showing that the utilization of irrelevant data has a nice theoretical justification. For clarity, we slightly modify Problem II to the following optimization problem. Based on the modified problem, we then derive the analysis. Problem II is changed as follows:

$$\begin{aligned}
\min_{\mathbf{w},b,\xi,\eta,\mathbf{y}_{l+1:n},\mathbf{d}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C_L \sum_{i=1}^{l} \xi_i \\
& + C_{rU} \sum_{j=l+1}^{n} \xi_j + C_{iU} \sum_{j=l+1}^{n} \eta_j \qquad (7.8) \\
\text{s.t.} \quad & y_i(\mathbf{w}_i \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \ldots, l \\
& y_j(\mathbf{w}_j \cdot \mathbf{x}_j + b) + \xi_j + M(1 - d_j) \geq 1, \\
& |\mathbf{w}_j \cdot \mathbf{x}_j + b| \leq \varepsilon + \eta_j + M d_j, \\
& d_j = \{0, 1\} \quad j = l + 1, \ldots, n. \\
& \eta_j \geq 0, j = l + 1, \ldots, n, \\
& \xi_k \geq 0, k = 1, \ldots, n.
\end{aligned}$$

C_{rU} represents the penalty parameter for the relevant samples, while C_{iU} describes the penalty imposed on the irrelevant data points. We first present the following theory.

Theorem 3. *The above learning machine with $C_{iU} = \infty$ and $\varepsilon = 0$ is equivalent to training a standard Transductive SVM [37]* *with the training points projected onto the orthogonal complement of span* $\{\mathbf{z}_j - \mathbf{z}_0, \mathbf{z}_j \in \mathcal{U}\}$*, where* $\mathbf{z}_0$ *is an arbitrary element of the space spanned by the irrelevant samples denoted by* $\mathcal{U}$*.*

Sketch of Proof: $C_{iU} = \infty$ and $\varepsilon = 0$ implies that any $\mathbf{w}$ yielding the optimal solution of (7.8) satisfies $\mathbf{w} \cdot \mathbf{z} + b = 0$ for any $\mathbf{z}$ judged as irrelevant samples. Hence, we have $\mathbf{w} \cdot (\mathbf{z} - \mathbf{z}_0) = 0$, implying $\mathbf{w}$ is orthogonal to the subspace spanned by all the irrelevant samples. Hence the optimization of (7.8) intends to find a traditional transductive SVM in a subspace which contains only the relevant samples, while the irrelevant samples are suppressed. In addition, from the previous argument, the space $\mathcal{U}$ spanned by the irrelevant samples can also benefit the classification, since it is $\mathcal{U}$ that decides the optimization subspace. □

Theorem 1 shows that the optimization of our proposed algorithm actually tries to find the most suitable subspace in which the margin can be maximized while the overall error can be minimized. The irrelevant data do not contribute to the final accuracy directly. However, it determines the subspace where the resultant decision boundary is derived and will consequently affect the final performance. Theorem 1 clearly shows how the irrelevant data can affect and eventually improve the overall performance.

Practical Solving Method

We now revisit the optimization of Problem II. Although there are softwares that are able to deal with MIP involved in Problem II, the computational complexity is usually high. It is even difficult to perform optimization with more than 50 $\{0, 1\}$ integer variables. Hence we would like to relax the problem to other solvable optimization forms. To achieve this purpose, we first reformulate Problem II to its dual form.

Problem III:

$$
\begin{aligned}
\max_{\lambda,\mathbf{z}^+,\mathbf{z}^-} \min_{\mathbf{y}_{l+1:n},\mathbf{d}} \quad & -\beta\top\mathbf{K}\beta + 2\sum_{i=1}^{n}\lambda_i - 2M\sum_{j=l+1}^{n}(1-d_j)\lambda_j \\
& -2M\sum_{j=l+1}^{n} d_j(z_j^- + z_j^+) \\
\text{s.t.} \quad & 0 \le \lambda_i \le C_L, i = 1,\dots,l && (7.9) \\
& 0 \le \lambda_j \le C_U, && (7.10) \\
& z_j^- + z_j^+ \le C_U, && (7.11) \\
& z_j^-, z_j^+ \ge 0, && (7.12) \\
& d_j = \{0,1\}, j = 1+1,\dots,n && (7.13)
\end{aligned}
$$

In the above, β_j is defined as $\beta_j = \begin{cases} \lambda_j y_j & j \le l \\ \lambda_j y_j + (z_j^- - z_j^+) & l+1 \le j \le n \end{cases}$. λ_j is the Lagrangian multiplier for (7.5) and (7.6) associated with $\mathbf{x}_j$, and z_j^- and z_j^- correspond the Lagrangian multipliers for (7.7) when the *abs* operator is expanded. And $\mathbf{K}$ is the kernel matrix defined as $\mathbf{K}_{i,j} = \mathbf{x}_i \cdot \mathbf{x}_j$.

Before proceeding to re-organized Problem III, we present some notation first. We denote a new vector $\alpha = (\lambda; \mathbf{z}_-; \mathbf{z}_+)$. We further define $\mathbf{P}_1 = (X\mathrm{Diag}(\mathbf{y}), X_{l+1:n}, -X_{l+1:n})\top$, where X represents the matrix $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$, $X_{k_1:k_2}$ represents the matrix consisting of the columns of X from k_1 to k_2, and $X \circ \mathrm{Diag}(\mathbf{y})$ represents the element-wise matrix multiplication of X and $\mathrm{Diag}(\mathbf{y})$. We further define $\mathbf{a} = (\mathbf{1}_l; \mathbf{1}_m - M(\mathbf{1}-\mathbf{d}); -M\mathbf{d}; -M\mathbf{d})$, where $\mathbf{1}_k$ represents a $\mathbf{k}$-dimension column vector with all the elements as 1. We denote the matrix $B = \begin{pmatrix} \mathbf{I}_{n\times n}, & \mathbf{0}_{n\times 2m} \\ \mathbf{0}_{m\times n}, & Q_{m\times 2m} \end{pmatrix}$, $Q_{m\times 2m} = (\mathbf{I}_{m\times m}, \mathbf{I}_{m\times m})$, $C = (\mathbf{C_L}_l; \mathbf{C_U}_{2m})$. Here $\mathbf{I}_{n\times n}$ is an $n \times n$ unit matrix, $\mathbf{0}_{k_1\times k_2}$ describes a $k_1 \times k_2$ matrix with all the elements as 0, and $\mathbf{C_L}_l$ defines an l-dimensional column vector with all the elements as C_L. Other symbols are similarly defined.

We can re-organized Problem III to the following problem by using the above notation.

$$\begin{aligned} \max_{\alpha} \quad \min_{\mathbf{y}_{l+1:n},\mathbf{d}} \quad & -\alpha\top\mathbf{P}_1\mathbf{P}_1\top\alpha + 2\mathbf{a}\top\alpha \\ \text{s.t.} \quad & \alpha \geq 0, \\ & B\alpha \leq C, \\ & d_j \in \{0,1\}, \forall j, l+1 \leq j \leq n. \end{aligned}$$

Once again, the dual form of the above optimization objective can be written to the following problem:

$$\max_{\alpha} \min_{\mathbf{d},\nu,\delta,\mathbf{y}_{l+1:n}} -\alpha\top\mathbf{P}_1\mathbf{P}_1\top\alpha + 2\alpha\top(\mathbf{a}+\nu) + 2\delta\top(\mathbf{C}-\mathbf{B}\alpha) \tag{7.14}$$

where $\nu, \delta \geq 0$ are the Lagrangian multipliers.

We can easily obtain the optimal $\alpha = (\mathbf{P}_1\mathbf{P}_1\top)^{-1}(\mathbf{a}+\nu-B\top\delta)$. Substituting the optimum value of α into (7.14), we further get the optimization problem as follows:

$$\begin{aligned} \max_{\alpha} \quad \min_{\mathbf{y}_{l+1:n},\mathbf{d},\nu,\delta} \quad & (\mathbf{a}+\nu-B\top\delta)\top(\mathbf{P}_1\mathbf{P}_1\top)^{-1}(\mathbf{a}+\nu-B\top\delta) + 2\delta\top\mathbf{C} \\ \text{s.t.} \quad & \nu \geq 0, \delta \geq 0, \\ & d_j \in \{0,1\}, \forall j, l+1 \leq j \leq n. \end{aligned}$$

Finally, the above optimization problem can equivalently be transformed to a form similar to the Semi-definite Problem (SDP) by using Schur Complement Lemma [82, 86].

Problem IV:

$$\begin{aligned} \min_{\mathbf{y}_{l+1:n},\mathbf{d},\nu,\delta,t} \quad & t \\ \text{s. t.} \quad & \begin{pmatrix} P & \mathbf{a}+\nu-B\top\delta \\ (\mathbf{a}+\nu-B\top\delta)\top & t-2\delta\top\mathbf{C} \end{pmatrix} \succeq 0, \\ & d_j \in \{0,1\}, \\ & y_j \in \{-1,+1\}, \forall j, l+1 \leq j \leq n. \end{aligned}$$

Here P is defined as

$$\begin{pmatrix} \mathbf{K}\circ(\mathbf{y}\mathbf{y}\top) & \mathrm{Diag}(\mathbf{y})\mathbf{K}_{1:n,l:n} & -\mathrm{Diag}(\mathbf{y})\mathbf{K}_{1:n,l:n} \\ \mathbf{K}_{1:n,l:n}\top\mathrm{Diag}(\mathbf{y}) & \mathbf{K}_{l+1:n,l+1:n} & -\mathbf{K}_{l+1:n,l+1:n} \\ -\mathbf{K}_{1:n,l:n}\top\mathrm{Diag}(\mathbf{y}) & -\mathbf{K}_{l+1:n,l+1:n} & \mathbf{K}_{l+1:n,l+1:n} \end{pmatrix}$$

and a matrix $\mathbf{A}\succeq 0$ means that $\mathbf{A}$ is a Semi-definite matrix.

Similar to the work presented in [86], we relax $(\mathbf{y}\mathbf{y}\top)$ as rank-one matrix $\mathbf{M}$. We further relax $d_j \in \{0,1\}$ to $0 \le d_j \le 1$. We can finally write the optimization problem as Problem V:

Problem V:

$$\begin{aligned} \min_{\mathbf{M},\mathbf{d},\nu,\delta,t} \quad & t \\ \text{s. t.} \quad & \begin{pmatrix} P & \mathbf{a}+\nu-B\top\delta \\ (\mathbf{a}+\nu-B\top\delta)\top & t-2\delta\top\mathbf{C} \end{pmatrix} \succeq 0, \\ & 0 \le d_j \le 1, \\ & rank(\mathbf{M}) = 1, \mathbf{M}_{1:l,1:l} = \mathbf{y}_{1:l}\mathbf{y}_{1:l}\top. \end{aligned}$$

Following most optimization methods in SSL [144, 148, 37, 137], we further remove the rank-one constraint, the above problem is exactly an SDP problem. Note that $\mathrm{Diag}(\mathbf{y})$ appearing in the matrix P can be represented by the elements of $\mathbf{M}$. For example, assume $y_1 = 1$, then $\mathrm{Diag}(\mathbf{y})$ can be written as $\mathrm{Diag}(M_{11}, M_{12}, ..., M_{1n})$. This SDP problem can be solved in polynomial time by some packages such as Sedumi[133].

7.2 Experiment

In this section, we evaluate our proposed framework on both synthetic and real data. A synthetic example will be firstly presented in order to illustrate the model clearly. We then compare our model with the traditional SSL and the Universum Support

Vector Machine (USVM) [140] on benchmark data sets, USPS [2] and MNIST data[3]. For brevity, we name our model as Universum Semi-supervised Learning, in short, USSL from now on. However, we should keep in mind that it is significantly different from the work presented in [157] in that the universum must be known beforehand in their work, while we do not have such requirement. Hence our proposed model presents a more general SSL framework. We implement our model by using a generic convex programming solver CVX.[4] The traditional SSL and the universum SVM are solved based on the package UniverSVM.[5]

7.2.1 Evaluation on Synthetic Data

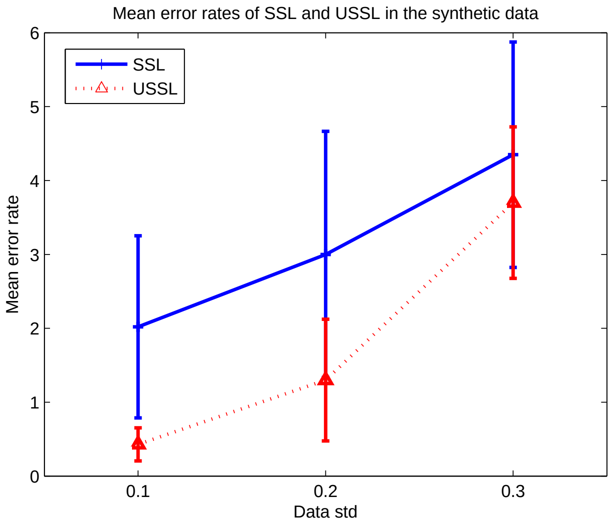


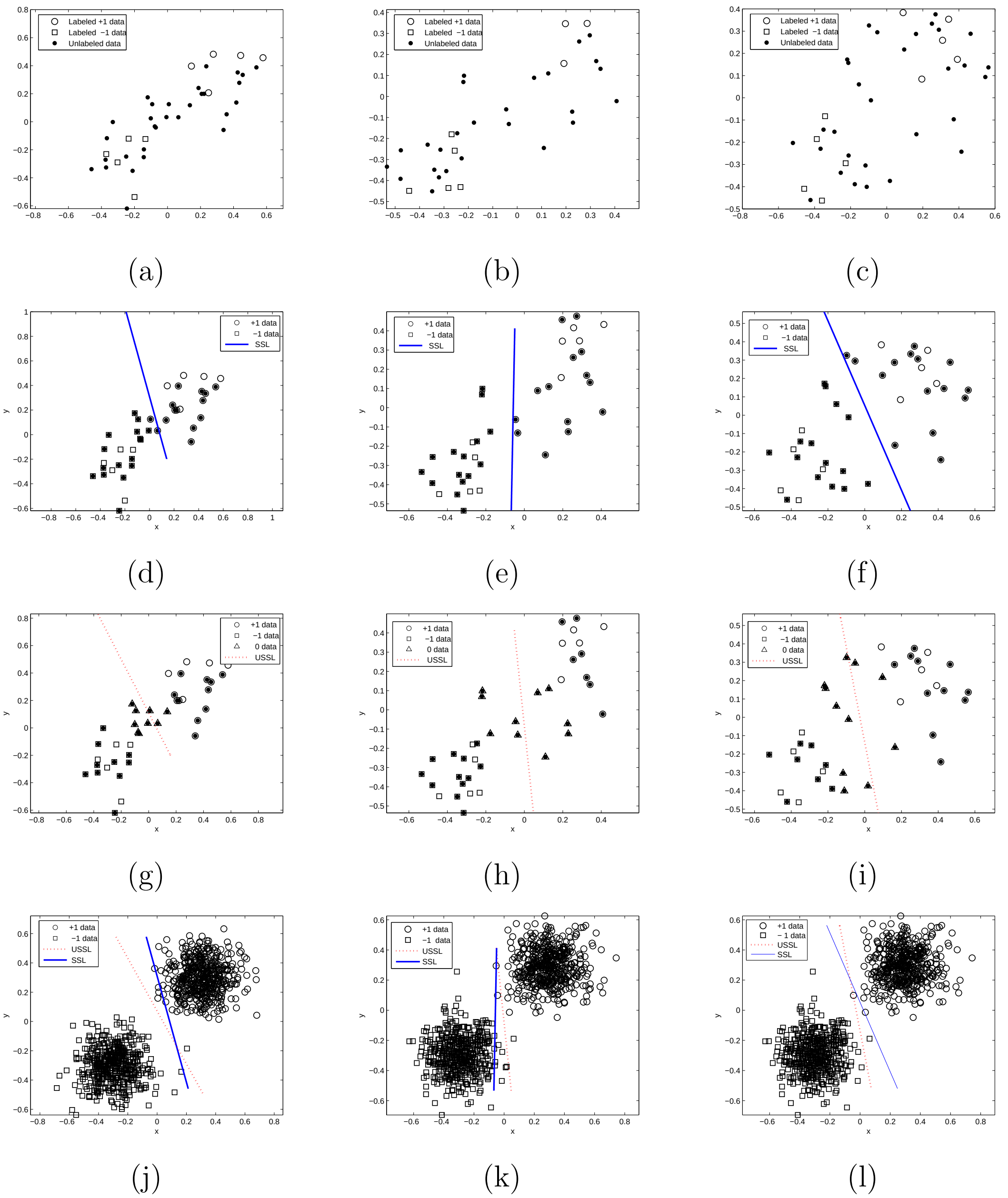
Figure 7.4: Comparison of SSL and the proposed USSL on the synthetic data

[2] The USPS data set can be downloaded from the web site http://www-stat-class.standford.edu / tibs/ElemStatLearn/data.html.

[3] The MNIST data set is available at http://yann.lecun.com/exdb/mnist.

[4] The matlab source codes of the CVX package can be downloaded from http://www.stanford.edu/ boyd/cvx/.

[5] The package of UniverSVM can be obtained from http://www.kyb.tuebingen.mpg.de/bs/people/fabee/universvm.html.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 7.5: Comparison of SSL and the proposed model USSL on synthetic data. (a)-(c) plot the training data for the three data sets respectively. (d)-(f) plot the decision boundary given by SSL as well as the class label of the unlabeled data assigned by SSL. (g)-(i) plot the decision boundary given by USSL as well as the class label of the unlabeled data assigned by USSL. (j)-(l) show the results on test data. The proposed USSL generates more reasonable decision boundaries and outperforms the traditional SSL.

We generate three synthetic data sets to validate our proposed algorithm. In more details, we obtain the training data for all the three data sets from three two-dimensional Gaussian distributions, which are centered at -0.3, 0, and $+0.3$ respectively. The two types of relevant data are centered at ± 0.3 both with the standard deviations as 0.13 for each data set, while the irrelevant data are located around 0, but with standard deviations as 0.1, 0.2, and 0.3 respectively for three data sets. The number of training samples for the labeled data and the relevant unlabeled data is respectively set to 5 and 30 for each class in all the three sets. The number of irrelevant unlabeled data samples for all the three cases is also set to 30. The test data consists of 500 samples for each class, generated from the same distributions as the labeled data. We train our proposed model USSL in comparison with SSL on the training data consisting of both irrelevant and relevant data samples, and evaluate its performance on the test data sets. In both SSL and USSL, C_U and C_L are set to 100. ε is set to 0.2. Note that again, we do not know which data samples are relevant or not beforehand. They are merely input as the unlabeled data for training in both USSL and SSL. The above process is repeated for 20 times and the average accuracy is reported in Figure 7.4.

It is obvious that the proposed general framework USSL demonstrates much better performance than SSL. The mean error rates of USSL are significantly lower than SSL in all the three cases. On the other hand, when the standard deviation increases, USSL tends to approximate the SSL in terms of the error rate, since it is difficult to detect irrelevant data in such cases.

In order to have a closer examination on the proposed USSL, we also draw the training set including the labeled and unlabeled data, the test data, and the decision boundaries for one of 20 evaluations in Figure 7.5. Figure 7.5(a), (b), and (c) show the training samples for the three sets, where the labeled samples are plotted as ◦'s and □'s for +1 and −1 class respectively, while ▪'s depict the unlabeled instances consisting of both relevant and irrelevant samples. Figure 7.5(d), (e), and (f) show the final class labels for the unlabeled data and the decision boundary given by the traditional SSL. The filled points represent the unlabeled data, but their shapes imply their class, i.e., the filled □'s are judged as −1 class, while the filled ◦'s are classified as +1 class. Similarly, we show the decision boundary given by USSL and the associated final class labels of the unlabeled samples for the three cases in Figure 7.5(g), (h), and (i) respectively. We use the similar symbols to describe different points. The difference is that our proposed USSL is able to indicate which samples are irrelevant. Such irrelevant samples are finally marked as ▲. It is interesting that almost all the irrelevant samples can be correctly detected by our proposed USSL as observed in these three sub figures. Moreover, the decision boundaries given by USSL are actually more reasonable than the ones derived by the traditional SSL. This can be also observed in Figure 7.5(j), (k), and (l), which show the test results for the three cases respectively.

7.2.2 Evaluation on Real Data

In this section, we evaluate the proposed novel model in comparison with the traditional SSL and the USVM [140] on real data, the USPS and the MNIST data. We follow [140, 126] and exploit the digits of 5 and 8 as the labeled data and use the remaining digits as the irrelevant data. Hence we have 8 data sets, depending on which category of digits is used as the irrelevant data. We randomly extract 20 labeled samples and 30 random data points as relevant unlabeled samples from 5 and 8 respectively. We further obtain 30 samples randomly extracted from a certain category of digits other than 5 and 8. The test data set contains 400 digits randomly extracted from the 5 and 8 digits. The parameters involved in SSL and USSL are searched via cross validation. More specifically, C_L and C_U are searched in $\{1, 10, 100, 1000\}$, while ε is searched in $\{0.1, 0.2, 0.3, 0.4\}$. The final test accuracy is given as the result averaged on the 10 random evaluations for both USPS and MNIST. In addition, as verified by many researches in Optical Character Recognition, especially in handwritten numeral recognition, kernel based methods are just slightly better than the linear classifier, but with significantly heavier computational cost.[6] Hence, we only conduct the comparisons based on the linear version of USVM, USSL and SSL in the following.

The evaluation results are reported in Table 7.1 and Table 7.2 for USPS and MNIST respectively. Once again, our proposed USSL outperforms the traditional SSL and the USVM. More specifically, the proposed USSL demonstrates significantly better performance than SSL and USVM in the 0, 1, 2, 3, 6, and 7 data sets of USPS according to a t-test at the 5% significance level. Similarly, a t-test indicates that the result of USSL is also significantly different from those of SSL and USVM in the 0, 1, 3, 4, 6, 7, and 9 data sets of MNIST at the significant level of 5%. SSL simply regards all the unlabeled data as relevant data, while USVM considers all the unlabeled data as universum. Hence it is inappropriate for them to deal with the general unlabeled data containing both relevant and irrelevant data. In comparison, our proposed approach can automatically model the impact caused by the relevant and irrelevant data into the final decision boundary. It demonstrates superior performance and is more appropriate in handling Semi-supervising Learning from general data.

[6] The performance of various methods on MNIST can be seen in the web site http://yann.lecun.com/exdb/mnist/.

Table 7.1: Experimental results on USPS data

Data set	USVM	SSL	USSL
0	67.05 ± 2.31	85.05 ± 1.94	$\mathbf{89.85 \pm 1.47}$
1	71.45 ± 1.59	83.61 ± 2.52	$\mathbf{89.23 \pm 1.89}$
2	69.50 ± 4.29	84.44 ± 2.08	$\mathbf{89.81 \pm 2.34}$
3	70.43 ± 1.68	84.75 ± 1.86	$\mathbf{89.65 \pm 2.24}$
4	65.80 ± 3.04	85.12 ± 3.91	$\mathbf{86.69 \pm 2.01}$
6	64.80 ± 2.36	78.45 ± 2.21	$\mathbf{83.70 \pm 1.90}$
7	66.93 ± 3.75	87.37 ± 2.51	$\mathbf{90.42 \pm 1.75}$
9	72.37 ± 3.42	82.86 ± 2.39	$\mathbf{85.13 \pm 2.31}$

Table 7.2: Experimental results on MNIST data

Data Set	USVM	SSL	USSL
0	45.25 ± 2.19	53.25 ± 2.84	$\mathbf{58.25 \pm 2.11}$
1	52.77 ± 1.42	54.10 ± 2.78	$\mathbf{60.25 \pm 2.75}$
2	54.58 ± 2.67	56.92 ± 3.12	$\mathbf{57.67 \pm 2.97}$
3	55.14 ± 1.90	52.09 ± 2.30	$\mathbf{57.25 \pm 1.32}$
4	56.65 ± 1.22	57.12 ± 2.49	$\mathbf{59.25 \pm 2.10}$
6	52.75 ± 2.80	54.50 ± 2.12	$\mathbf{57.67 \pm 1.27}$
7	60.51 ± 2.12	58.09 ± 3.01	$\mathbf{68.50 \pm 2.26}$
9	59.25 ± 1.15	48.25 ± 2.64	$\mathbf{63.00 \pm 1.50}$

7.3 Summary

We have proposed a novel framework that can learn from general unlabeled data. In contrast to the traditional Semi-supervised Learning that requires unlabeled data to share the same category labels as the labeled data, the proposed framework is able to learn from unlabeled data with irrelevant samples. Moreover, we do not need the prior knowledge on which data samples are relevant or irrelevant. Consequently it is significantly different from the recent Semi-supervised Learning with universum or the Universum Support Vector Machines. As an important contribution, we have successfully formulated this new learning approach as a Semi-definite Programming problem, making it solvable in polynomial time. We have also presented theoretical analysis to justify our model. A series of experiments demonstrate that this novel framework has advantages over the Semi-supervised Learning on both synthetic and real data in many facets.

□ **End of chapter.**

Chapter 8

Semi-supervised Learning by Active Search

We consider the application of semi-supervised learning in the text categorization area. Automated text categorization, which is a fundamental step toward text and web mining applications, has become an important subject in both the research and application communities. The goal of automated text categorization is to automatically classify documents into predefined categories. It is regarded as a supervised learning problem where a statistical classification model is learned from a pool of labeled documents. Since the performance of statistical classifiers often depends on the availability of labeled examples, one of the major bottlenecks toward automated text categorization is to collect sufficient numbers of labeled documents because of the high cost in manually labeling documents. For example, there are often only a few blog documents in a blog website that have been manually categorized according to their blog types or topics (e.g., nursing, nutrition and etc). Given a small number of labeled documents, it is very challenging, if not impossible, to build a reliable classifier that is able to achieve high classification accuracy.

One way to address the problem of small-size sample is to exploit the unlabeled documents by so-called semi-supervised learning methods. There are two major groups of approaches toward semi-supervised learning. The first group of approaches is based on the clustering assumption which assumes that most documents, including both the labeled ones and the unlabeled ones, should be far away from the decision boundary of the target classes. The typical approaches in this category include Tranductive SVM (TSVM) [75, 145, 148] and Semi-supervised SVM (Semi-SVM) [34, 37]. The second group of approaches is based on the manifold assumption which assumes that most documents lie on a low-dimensional manifold in the input space. The well-known algorithms in this category include Label Propagation [160], Markov random field [164, 46], Graph Cuts [18], and Spectral Graph Transducer [76]. A comprehensive study of semi-supervised learning techniques can be found in [163, 28]. In order to exploit the semi-supervised learning techniques, one of the major issues is to obtain a multitude of unlabeled documents that are relevant to the target categories.

One way to collect the unlabeled documents is through the Web search engines. In order to retrieve Web documents that are relevant to the target topics, we will first identify the keywords from a few labeled documents that are closely related to the target topics. Web documents will then be retrieved by the Web search engine using the textual queries that are constructed based on the identified keywords. Finally, the retrieved Web documents will be combined with the labeled documents to construct a text classification model using the semi-supervised learning techniques. We refer to this framework as "**Semi-supervised Text Categorization by Active Search**", whose goal is to enhance a text classification model by actively exploiting the unlabeled Web documents via the Web search engines. Figure 8.1 illustrates the key features of this framework, i.e.

1. *Query generation* that generates the textual queries for document retrieval by analyzing the content of labeled docu-

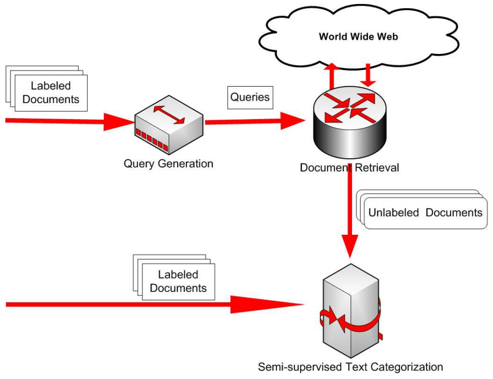


Figure 8.1: The framework of semi-supervised text categorization by active search.

ments,

2. *Document retrieval* that retrieves the Web documents through the Web search engine by using the generated queries, and

3. *Semi-supervised text categorization* that constructs text classification models by utilizing both the labeled documents and the unlabeled Web documents which are retrieved by the Web search engine.

Our approach is motivated by the observation that people usually utilize Web search engines such as Google and Yahoo! to search useful documents when they are unclear about certain topics/concepts. The Web pages or documents returned by Web search engines usually help users better understand the target concepts if the query terms are carefully chosen. This fact motivates us to collect unlabeled documents that are relevant to the target categories via the Web search engines. The retrieved Web documents will be combined with a few labeled documents to build more reliable text classification models. This situation is also analogous to a user-expert model that the user can strengthen his knowledge by asking experts appropriate questions. The Web comprises of a huge number of Web pages and documents that cover a wide range of topics. It can be viewed as an expert of almost any field. Ideally, if we can properly design questions and understand the answers retrieved from the Web, we should be able to correctly classify the topics of new documents even with a few labeled examples. Since the process of asking questions and getting the answer is automatically conducted without user interactions, we refer to this novel Web-assisted classification scheme as "**Semi-supervised Text Categorization by Active Search**."

We discuss the details of the semi-supervised learning framework proposed in [147]. Our work present a novel framework of actively retrieving related documents from the Web as a complementary information source for supervised text categorization. It should be noted that the Web is also used as the complementary information for other tasks such as author resolution identity [78, 110]. A similar idea also appears in the empirical study of [54, 156], where the preliminary results indicate the usefulness of the information from the Web. Our work is also distinguished from search engine based methods [1] where the search engines are used as the feature selection tools and a large corpus of same-category documents are available. It is also interesting to note the relationship and the difference among the proposed framework, active learning and transfer learning. Both the proposed framework and active learning [97, 58] aim to actively select unlabeled examples to improve the results of supervised learning. However, they differ in that the examples selected by active learning will be manually labeled and used to augment the pool of training examples. In contrast, the unlabeled document collected by the proposed framework will remain unlabeled. Moreover, the proposed framework is similar to transfer learning or domain adaption [25, 2, 43, 73, 52, 112] in that both of them aim to transfer knowledge from some domain to the classification models of the target concepts. However, the key difference between the proposed framework and transfer learning is that transfer learning assumes the availability of the examples (both labeled and unlabeled) from different but related topics/domains while the proposed framework does not. Indeed, one of the key components within the framework is how to collect relevant unlabeled documents from the Web via the Web search engines.

As a key contribution, we present a novel learning approach, named Discriminative Query Generation (DQG), for query generation that improves the chance of finding the documents relevant to the target topics via Web retrieval. Both theoretical justifications and empirical evaluations demonstrate that the DQG approach significantly outperforms other intuitive methods such as Term Frequency (TF) [53], Term Frequency/Inverse Document Frequency (TF/IDF) [20], and Odds-ratio [53]. Furthermore, we engage the semi-supervised learning method to perform text categorization that can effectively exploit both the labeled documents and the unlabeled Web documents which are retrieved by Web search engines. Extensive results show that semi-supervised learning framework is consistently superior to the purely supervised text categorization method.

The remaining of this chapter is organized as follows. In section 8.1, we describe the framework of semi-supervised text categorization by active search, and algorithms for query generation and semi-supervised text categorization. Section 8.2 presents the empirical study of the proposed text categorization framework. We draw conclusions of this study in Section 8.3.

8.1 Semi-supervised Text Categorization by Active Search

In this section, we will first present the problem of semi-supervised text categorization by active search, followed by the description of the novel algorithms for the two key components of our proposed text categorization framework, i.e., query generation and semi-supervised text categorization methods.

8.1.1 Problem Definition

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \dots, n_l\}$ denote the collection of labeled documents where $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of d dimension that represents the content of the ith document and $y_i \in \{-1, +1\}$ is a binary class label assigned to the ith document. In addition to the collection of labeled documents $\mathcal{D}$, we assume that there is another much larger collection of documents, denoted by $\mathcal{U}$, that can only be accessed through the search engine $\mathcal{A}$. This large collection $\mathcal{U}$ can either be a collection of Web pages that are accessible via the Web search engine, or a collection of biological research articles that are accessible via the PubMed. We abstract the search engine $\mathcal{A}$ as a ranking function that takes a textual query[1] $\mathbf{q} \in \{0, 1\}^d$ as an input and outputs a ranking list of documents in $\mathcal{U}$. To make the problem more practical, we assume that only the first s documents in the ranking list are accessible. The goal of the proposed semi-supervised text categorization framework is to learn a text classification model that exploits both the labeled documents $\mathcal{D}$ and the unlabeled documents in $\mathcal{U}$ via the search engine $\mathcal{A}$.

As already shown in the introduction section, we divide the procedure of the proposed text categorization framework into three steps, namely (1) query generation, (2) document retrieval, and (3) semi-supervised text categorization. In the following subsections, we will focus on the discussion of query generation and semi-supervised text categorization since the step of document retrieval is naturally taken care of by the given search engine $\mathcal{A}$.

[1] Note that each query vector is a binary vector.

8.1.2 Discriminative Query Generation (DQG)

Given the collection of labeled documents $\mathcal{D}$, the goal of query generation is to construct a set of queries $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_t\}$ that are likely to retrieve documents relevant to the target concepts. Since each query $\mathbf{q} \in \{0, 1\}^d$ is a binary vector, query generation essentially is to decide which words will be included into the query. Thus, we can view the problem of query generation as a feature selection problem, i.e., selecting the subset of word features that are most representative for the given classification task. A straightforward approach to employ the common statistical measurements, such as TF, TF/IDF, information gain, and χ^2 statistic, to select the most informative word features. However, given the limited number of training documents, it is unlikely to obtain reliable estimates for any statistical measurements. As a result, the selected word features may not necessarily be the most representative for the target classification task. We refer to this problem as the "**sparse data**" problem for query generation. Another drawback with the feature selection approaches is that even if the selected word features are representative for the given task of text categorization, they may be completely unrelated since they may be extracted from different documents. As a result, by having a query that consists of multiple unrelated words, the search engine $\mathcal{A}$ is likely to return a few even none documents. We refer to this problem as the "**unrelated query words**" problem for query generation.

To address the problem of unrelated query words, we propose to generate a query for every labeled document. Let $\mathbf{q}_i$ be a query that we will generate from the document $\mathbf{x}_i$. For the convenience of discussion, we assume that $\mathbf{x}_i$ is a positively labeled document, i.e., $y_i = +1$. Let V_i denote the vocabulary used by document $\mathbf{x}_i$, i.e., $V_i = \{k | x_{i,k} > 0\}$. We first restrict the words used by query $\mathbf{q}_i$ to vocabulary V_i, i.e., $q_{i,k} = 0$ if $k \notin V_i$. To measure the informativeness of words, we introduce a non-negative weight $w_i \geq 0$ for each word. The more informative a word is, the larger the weight will be. We follow the framework of Support Vector Machine (SVM) and determine the word weights by the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{w},\xi} \quad & \sum_{j \in V_i} w_j + C \sum_{k=1}^{n_l} \xi_k \qquad (8.1)\\
\text{s. t.} \quad & y_k \left(\sum_{j \in V_i} w_j x_{k,j} + b \right) \geq 1 - \xi_k, \xi_k \geq 0, k = 1, \ldots, n_l \ ,\\
& w_j \geq 0, \ \forall j \ ,\\
& w_j = 0, \ \forall j \notin V_i.
\end{aligned}$$

where C is the parameter that weights between the classification error $\sum_{k=1}^{n_l} \xi_k$ and the regularization term $\sum_{j \in V_i} w_j$, and b is the bias. The word features with the largest weights will be selected to form a query. Note that according to the statistical learning theory [138], by introducing the regularization term, we should be able to reduce the mistakes in identifying informative word features that may be caused by the sparse data problem.

In order to see which words will be assigned with large weights, we rewrite the objective function as[2]

$$\mathcal{L} = \sum_{j \in V_i} w_j + C \sum_{k=1}^{n_l} \max \left(0, 1 - y_k \left(\sum_{j \in V_i} w_j x_{k,j} - b \right) \right) .$$

The subgradient [22][3] of $\mathcal{L}$ is then written as

$$\frac{\partial \mathcal{L}}{\partial w_j} = \left\{ \begin{array}{cc} 0 & j \notin V_i \\ & \\ 1 + \sum_{k=1}^{n_l} \eta_k & j \in V_i \end{array} \right. ,$$

[2] Here, we simplify the discussion by ignoring the constraint $w_j \geq 0, \forall j$

[3] Note that here we can only define the subgradient for $\mathcal{L}$ because of the non-smooth function max in $\mathcal{L}$

where η_k is the sub-derivative of $\max(0, 1 - y_k(\sum_{j\in V_i} w_j x_{k,j} - b))$ and is expressed as follows:

$$\eta_k = \left\{ \begin{array}{cc} 0 & y_k(\sum_{j\in V_i} w_j x_{k,j} - b) > 1 \\ -y_k x_{k,j} & y_k(\sum_{j\in V_i} w_j x_{k,j} - b) < 1 \\ (0, -y_k x_{k,j}) & y_k(\sum_{j\in V_i} w_j x_{k,j} - b) = 1 \end{array} \right. .$$

Since the large weights w_j tend to be assigned to the word features that have the most negative sub-derivatives $\partial\mathcal{L}/\partial w_j$, we expect the words to have large weights if they are mainly used by the positively labeled documents, i.e., $y_k x_{k,j} > 0$ if $y_k = +1$ and $y_k x_{k,j} = 0$ if $y_k = -1$. This is clearly consistent with our intuition.

For a negatively labeled document $\mathbf{x}_i$, we will have a similar optimization problem:

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \sum_{j\in V_i} w_j + C\sum_{k=1}^{n_l} \xi_k \\ \text{s. t.} \quad & -y_k\left(\sum_{j\in V_i} w_j x_{k,j} - b\right) \geq 1 - \xi_k, \xi_k \geq 0, k = 1, \ldots, n_l, \\ & w_j \geq 0, \ \forall j, \\ & w_j = 0, \ \forall j \notin V_i. \end{aligned}$$

Finally, for the document $\mathbf{x}_i$, the query $\mathbf{q}_i$ will be composed by the top k words with largest values of $\mathbf{w}$. It is important to note that although we discuss the query generation problem for the binary data, it is easy to generalize to the problem of generating query words for multi-class documents using approaches such as one-against-others. In order to differentiate from the existing approaches for query generation, we refer to the proposed method as the "**discriminative query generation**" method, or **DQG** for short.

8.1.3 Text Categorization Methods

Given the labeled documents $\mathcal{D}$ and the collection of unlabeled Web documents $\mathcal{U}$ returned by the search engine $\mathcal{A}$, the next question is how to construct a binary classification model $h(\mathbf{x}) : \mathcal{X} \rightarrow \{-1, +1\}$ for text categorization that exploits both the labeled and the unlabeled documents. In this subsection, we will examine several learning techniques for semi-supervised text categorization.

Given a query $\mathbf{q}_i$ that is generated from the labeled document $(\mathbf{x}_i, y_i)$, we denote by $\mathcal{U}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)})$ the collection of documents retrieved by the search engine $\mathcal{A}$, where n_i is the number of retrieved documents. Since the retrieved documents are closely related to the query $\mathbf{q}_i$, we would expect that the class labels assigned to the documents in $\mathcal{U}^{(i)}$ should also be closely related to y_i, i.e., the class label assigned to $\mathbf{x}_i$. Based on whether or not this assumption holds, we present two different approaches for text categorization that combine both the labeled documents and the unlabeled retrieved documents:

- The *auxiliary approach* that assumes all the retrieved documents in $\mathcal{U}^{(i)}$ belong to the category y_i.
- The *semi-supervised learning approach* that does not assume any relationship between the class labels assigned to $\mathcal{U}^{(i)}$ and the class label y_i.

The following subsections provide detailed description for these two approaches.

Auxiliary Approach

Most learning algorithms can be regarded as finding a classification model $h(\mathbf{x})$ that minimizes a certain loss function $L(h(\mathbf{x}), y)$ defined on the predicted category $h(\mathbf{x})$ and the labeled category

y. Similar to Section 8.1.2, we assume y to be a binary category, i.e., either -1 or $+1$. In addition to the empirical loss function $L(h(\mathbf{x}), y)$, to avoid the overfitting problem, a regularization term, denoted by $F(h)$, is often introduced to control the complexity of the classification model $h(\mathbf{x})$. Combining these two terms, we have the following objective function for learning classification model $h(\mathbf{x})$.

$$J(h) = \sum_i L(h(\mathbf{x}_i), y_i) + \lambda F(h) \ ,$$

where λ is a tradeoff parameter that balance the empirical loss function against the regularization term.

In the auxiliary approach, we assume that the documents retrieved by query $\mathbf{q}_i$ share the same class labels as y_i, i.e., the class assigned to document $\mathbf{x}_i$. This assumption allows us to label the retrieved Web documents that are originally unlabeled, which we refer to as "*auxiliary labeled documents*". We will train a classification model using both the labeled documents and the auxiliary labeled documents. This general idea leads to the following objective function:

$$J_a(h) = \sum_{\mathbf{x}_i \in \mathcal{D}} L(h(\mathbf{x}_i), y_i) + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} L(h(\mathbf{x}_j), y_j^*) + \lambda F(h) \ .$$

In the above, we introduce $\mathcal{U}$ to represent the collection of the auxiliary labeled documents, and y_j^* represents the class label of the retrieved Web document $\mathbf{x}_j$ that is predicted by the method mentioned above. Parameter $0 \leq \gamma \leq 1$ is introduced to weight the classification errors related to the auxiliary labeled documents. By setting γ to be less than 1, the objective function is more tolerable with the classification mistakes of the auxiliary labeled documents than with the labeled documents.

To further simplify our computation, we assume $h(\mathbf{x})$ to be a linear function[4], i.e., $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$. Following the frame-

[4] Previous studies have shown that linear classifiers are usually more effective than

work of Support Vector Machines (SVM), we define $F(h)$ as the L_2-norm $\|\mathbf{w}\|_2^2$, and a hinge loss function for $L(h(\mathbf{x}_i), y_i)$, i.e., $L(h(\mathbf{x}_i), y_i) = \max(0, 1 - h(\mathbf{x}_i)y_i)$. We thus have the following concrete optimization problem for text categorization that learns from both the labeled documents and the auxiliary labeled documents:

$$\begin{aligned} \arg\min_{\mathbf{w},b} \quad & \lambda\|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i \in \mathcal{D}} \xi_i + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} \xi_j \\ \text{s. t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i \ \mathbf{x}_i \in \mathcal{D} \ , \\ & y_j^*(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \xi_j, \quad \forall j \ \mathbf{x}_j \in \mathcal{U} \ . \end{aligned} \tag{8.2}$$

The above optimization problem can be efficiently solved by the standard quadratic programming packages.

The main shortcoming with the proposed auxiliary approach is the assumption that the Web retrieved documents are completely relevant to the generated query, and therefore share the same category as that of the labeled document used to generate the query. Due to the inaccuracy of the query generation method and the mistakes made by the search engines, it is likely that some of the retrieved Web documents are irrelevant to the generated queries and therefore are unrelated to the target text categories. To address this problem, we present a semi-supervised learning framework that combines the labeled documents and the unlabeled documents retrieved from the Web when constructing text categorization models.

Semi-supervised Approach

In the semi-supervised approach to text categorization, we do not assume that the retrieved documents are completely relevant to the generated queries, and therefore no assumption is made for the class labels of the documents retrieved from the Web.

nonlinear classifiers for text categorization

In this approach, the semi-supervised SVM is used to exploit both labeled documents and the Web retrieved documents for building text categorization models. Semi-supervised SVM tries to maximize the margin in the presence of unlabeled data and learns a decision boundary that traverses through low density regions while respecting the labels in the input space [34]. Different from the auxiliary approach, the semi-supervised learning approach optimizes not only the classification function $h(\mathbf{x})$, but also the class label assigned to the unlabeled data $\mathbf{y}^*$. The overall objective function used by semi-supervised SVM is formulated as follows:

$$J_s(h, \mathbf{y}^*) = \sum_{\mathbf{x}_i \in \mathcal{D}} L(h(\mathbf{x}_i), y_i) + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} L(h(\mathbf{x}_j), y_j^*) + \lambda F(h) \ ,$$

where γ and λ are similar to the ones defined as in the auxiliary approach. Note the key difference between J_s and J_a is that both class labels $\mathbf{y}^*$ and classification model $h(\mathbf{x})$ are optimization variables in J_s while only $h(\mathbf{x})$ is unknown variable in J_a. The related optimization problem is formulated as follows:

$$\begin{aligned} \arg\min_{\mathbf{w}, b, \mathbf{y}^*} \quad & \lambda \|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i \in \mathcal{D}} \xi_i + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} \xi_j \ , \\ \text{s. t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i \ \mathbf{x}_i \in \mathcal{D} \ , \\ & y_j^*(\mathbf{w}^\top \mathbf{x}_j + b) \geq 1 - \xi_j, \quad \forall j \ \mathbf{x}_j \in \mathcal{U} \ . \end{aligned} \tag{8.3}$$

Similar to the auxiliary approach, we use $\|\mathbf{w}\|_2^2$ for $F(h)$ and $L(h(\mathbf{x}_i), y_i) = \max(0, 1 - h(\mathbf{x}_i) y_i)$, the objective function is simplified as follows:

$$\begin{aligned} J_s(h) \ &= \ \lambda \|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i \in \mathcal{D}} + \max(0, 1 - h(\mathbf{x}_i) y_i) \\ & \quad + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} \max(0, 1 - |h(\mathbf{x}_j)|) \ . \end{aligned}$$

Note that variables y_i^* is removed because

$$\min_{y_j^* \in \{-1,+1\}} \max(0, 1 - h(\mathbf{x}_j) y_j^*) = \max(0, 1 - |h(\mathbf{x}_j)|) \ .$$

The above formulation is not convex because the function $\max(0, 1 - |h(\mathbf{x}_j)|)$ is neither convex nor concave. Several approaches have been proposed to address this computational problem (see [28] for reference.) A recently proposed semi-supervised learning technique is the Concave-Convex Procedure (CCCP) [37] which efficiently solves the semi-supervised SVM and can be used for large scale semi-supervised learning problems. The key idea is to rewrite the loss function of unlabeled data into the difference of two convex functions. In particular, in the CCCP solution to semi-supervised SVM, a so-called Ramp loss L_s is introduced such that the loss function for an unlabeled data point $\mathbf{x}_j \in \mathcal{U}$ is written as a sum of a convex function and a concave function. This is indeed equivalent to replacing the symmetric Hinge loss $L_1(|h(\mathbf{x}_j)|)$ with the Ramp loss and in the meantime, duplicating each unlabeled example into two copies with one assigned to the positive class and the other assigned to the negative class. It amounts to the following objective function:

$$\begin{aligned} J_s(h) &= \lambda \|\mathbf{w}\|_2^2 + \sum_{\mathbf{x}_i \in \mathcal{D}} + \max(0, 1 - h(\mathbf{x}_i) y_i) \\ &\quad + \gamma \sum_{\mathbf{x}_j \in \mathcal{U}} \left(L_s(h(\mathbf{x}_j), +1) + L_s(h(\mathbf{x}_j), -1) \right) \ . \end{aligned}$$

A CCCP procedure can be directly applied to optimize the above problem.

8.2 Experiment

In this section, we evaluate our proposed semi-supervised text categorization framework. We first describe the data sets that are used for evaluation, followed by the setup of the experiments. We then present evaluation results for the key components of the text categorization framework, i.e., the query generation method and text categorization methods, in details.

8.2.1 Data Set

Nine subsets of text documents are selected from three benchmark text collections, including 20 Newsgroups, Reuters-21578, and Ohsumed.The 20 Newsgroups data set is a collection of approximately $20,000$ newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. The Reuters-21578 corpus is a collection of documents that appeared on Reuters newswire in 1987. The Ohsumed database is a on-line medical information database, consisting of titles and/or abstracts from medical journals. The description of the selected nine subsets can be found in Table 8.1, where *fourDiseases* is composed by the 7th to 10th categories of Ohsumed, and *sci* is composed by four news groups under the sci domain. The binary data sets are named according to their class labels. We select the data sets based on their semantic connections among the category labels in order to make them difficult to classify.

As shown in Figure 8.1, the proposed text categorization framework is composed of three major steps, namely (1) query generation, (2) document retrieval, and (3) text categorization. In the following subsections, we will focus on evaluating the query generation methods and text categorization methods that exploit both labeled documents and the unlabeled ones retrieved by search engines. In addition, we also briefly evaluate the classification accuracy of our proposed framework given different

Table 8.1: Data sets for evaluation.

Corpus	**Data set**	**# Docs**	**# Classes**
Newsgroup	auto vs. motor	2000	2
	sci	4000	4
Ohsumed	musculo vs. digestive	772	2
	bacterial vs. virus	581	2
	male vs. female	871	2
	fourDiseases	1319	4
Reuters	corn vs. wheat	520	2
	ship vs. trade	772	2
	money vs. trade	1203	2

search engines.

8.2.2 Evaluation (I): Query Generation

To evaluate queries generated from a limited number of labeled documents, we try several query generation methods, including Term-frequency (TF), Term Frequency/Inverted Document Frequency (TF/IDF), SVM with largest weights (shorted as SVM-LW), Odds Ratio (OR), and our Discriminative Query Generation (DQG) method. Different from the other methods in comparison, DQG generates a batch of queries with one query for each labeled document.

Unfortunately, there is no existing data set which is available for evaluating the query generation methods. Since our goal is to improve the performance of text categorization by retrieving relevant documents from the Web, we can evaluate the query generation methods indirectly by measuring the classification accuracy of text categorization using the Web documents that are retrieved by different query generation methods. The drawback of this method is that the evaluation is indirect and the classification accuracy is influenced by both query generation methods and text categorization methods. In order to directly evaluate the query generation methods, we will examine the following two different aspects of query generation methods:

- *Relevance*, i.e., whether or not the query generation method is able to identify the keywords that are representative to the target categories, and
- *Discriminativeness*, i.e., whether or not the generated query is able to differentiate documents from different categories.

To examine these two properties, we manually check the queries generated by different query generation methods. For the limit of space, we take two text categorization tasks (*bacterial vs. virus* and *money vs. trade*) as an example. For each task, the query terms are generated from 10 randomly chosen documents and the remaining documents with the query terms as the feature set are used as the test data. In Table 8.2 and Table 8.3, we list the first three query words that are chosen by five query generation methods for these two tasks. We also list the classification accuracy of SVM on the test data for each task. Note that since our proposed method generates a batch of queries, in order to compare with other query generation methods that generate only one query for the training documents, we extract the most frequent three query words from all the queries generated by the proposed method.

Since the *bacterial vs. virus* data set consists of paper abstracts about the bacterial disease and the virus disease, the query words should represent documents in these two categories and also distinguish the bacterial disease from the virus disease. We observed that the TF method only finds the frequent terms, but cannot find terms that can differentiate documents of the two diseases. For example, the term “patient” is selected by the TF method for both categories, and is therefore not able to tell documents of two diseases apart. Other methods, such as

TF/IDF and OR, are prone to finding the terms that may be irrelevant to the target categories. Examples of irrelevant terms are "percentage", "role", and "clear". The failure of TF/IDF and OR can be explained by the small number of documents that are used for selecting query words. Although both TF/IDF and OR have taken into account the factor of discriminativeness when selecting query words, due to the limited number of samples, these statistics cannot be measured accurately. As a result, inappropriate query words are chosen by these two methods. In contrast, the proposed DQG method is able to select query words that are not only relevant to the target categories and but also discriminative enough to distinguish different text categories. For example, "antibody" is directly related to the virus disease while "pneumonia" and "organism" are highly related to the bacterial disease. A similar analysis is applied to the *money vs. trade* data set. It is interesting to note that although the terms "said" and "quarter" seem to be irrelevant to the document categories of money and trade, they are indeed very useful to locate specific documents from the Web. Because "said" and "quarter" are commonly used by news documents, they are effective in retrieving news pages since the data set *money vs. trade* is a part of the Reuters news collection. In addition, in both cases, the prediction accuracy on the test data sets shows that the proposed DQG works consistently better than other query generation methods. Therefore, our proposed DQG method appears to be more effective in identifying query words that are both representative and discriminative for the target categories.

In order to quantitatively evaluate the query generation methods, we employ the information retrieval task and evaluate the quality of the generated queries by measuring the percentage of the retrieved documents that share the same category as the documents used to generate queries. To this end, we first construct a test database using the existing text corpus. The Indri search engine from the Lemur project (www.lemurproject.org/) is employed to build indices for the database.

In order to compare with other query generation methods, we merge the multiple queries generated by the proposed DQG method into single one by extracting the top N popular terms from the multiple generated queries. For each data set (i.e., *musculo vs. digestive* and *coin vs. wheat*), we randomly select 10 labeled documents with 5 documents for each category, and generate the queries based on the selected documents. We also vary the length of generated queries from 3 to 5 in order to obtain a full spectrum of performance evaluation. For each generated query, we compute the precision (i.e., percentage) of the top 100 retrieved documents that share the same category as the documents used to generate query. To avoid the effect from variance, each experiment is repeated independently five times. The results are shown in Table 8.4. It can be observed that our proposed DQG method demonstrates overall better performance than the other methods. For the *musculo vs. digestive* data set, DQG consistently outperforms the other methods; for the *coin vs. wheat* data set, DQG ranks the first when the query size is set to be 3, and ranks both the second when the number of query words is increased to 4 and 5. We thus conclude that DQG is overall a better approach for query generation. In the following study, we will always use DQG for query generation. Since DQG tends to produce its best performance on average when the query size is 3, in the following experiments, we always set the query size to be 3 for each labeled document, which is also supported by the empirical study in [156].

8.2.3 Evaluation (II): Text Categorization Methods

In this subsection, we describe the experimental settings, followed by the presentation of experimental results.

Experimental Setup

We employ SVM, which is regarded as the state-of-the-art classification technique, as our baseline algorithm. The SVM-light (http://svmlight.joachims.org/) implementation is utilized. Two learning algorithms are implemented for text categorization, including the auxiliary SVM (abbreviated as Auxi-SVM) and semi-supervised SVM (abbreviated as Semi-SVM). As described before, both algorithms are able to utilize the unlabeled documents retrieved from the Web for text categorization. For semi-SVM, the concave-convex procedure is implemented for efficient optimization.

We conduct two sets of experiments. In the first set of experiments, we randomly select 2 labeled documents per class of data to form the training set for each data set, and use the remaining documents as the test set. In the second set of experiments, 5 labeled documents[5] per class are randomly selected to form the training set. For each document, one query of three terms is generated to retrieve similar documents from the Web. We download the first 100 documents returned by each query based on the assumption that most search engines rank relevant documents before the irrelevant ones. Thus, for a binary data set, given a training set that contains 10 labeled documents, a total of $1,000$ unlabeled documents will be downloaded. In both sets of experiments, Google is used as the Web search engine for retrieving Web documents.

We represent each document by a vector of term frequency.

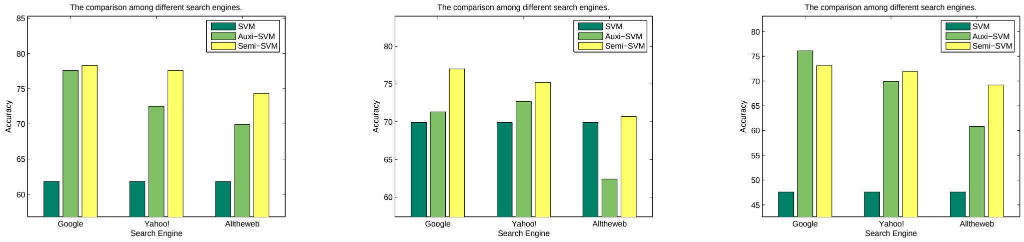
[5] For the sake of consistence, the 10 labeled documents include the documents selected by the first set of experiments.

For all the data sets used for experiments, the term frequency vectors of documents are normalized to be one. We then select 600 most informative features according to their correlation to the text categories. Given the limited number of labeled documents, it is often difficult to tune parameters γ and λ. The parameter λ is set to 0.01 for all the data sets. γ is also set to 0.01 in order to emphasize the importance of the margin error on the labeled data.

Experimental Results

Table 8.5 summarizes the classification accuracy of the supervised classification method (SVM) and the two methods (i.e., Auxi-SVM, and Semi-SVM) which utilize the retrieved data for the first and second sets of experiments. For an easy comparison, we also list the average accuracy of each method at the end of the table. When the training size per class is equal to 2, both the categorization methods using the retrieved documents achieve significantly higher accuracy than the supervised method for almost all data sets. On average, SVM achieves an accuracy of 53.1%, while the classification accuracy is 61.4% for Auxi-SVM and 66.3% for Semi-SVM. The overall reduction in classification error for Auxi-SVM is 17.6% and 28.0% for Semi-SVM. A similar observation is found when the training size per class is increased to 5: the average classification accuracy is increased from 61.1% to 71.3% and 74.3% when using the documents retrieved from the Internet. The overall error reduction in this case is 26.3% for Auxi-SVM and 34.0% for Semi-SVM. In conclusion, for both cases, using the documents retrieved from the Internet can greatly improve the classification accuracy. The advantage is more manifest when the semi-supervised SVM is adopted for text categorization.

8.2.4 Evaluation (III): Impact of Search Engines



(a) *bacterial vs. virus* (b) *musculo vs. digestive* (c) *male vs. female*

Figure 8.2: The classification accuracy of text categorization methods (i.e., Auxi-SVM and Semi-SVM) using different search engines (i.e., Google, Yahoo!, and Alltheweb) on three data sets of Ohmued

In this experiment, we aim to evaluate the impact of search engines on the classification accuracy of text categorization. Three Web search engines are used in this study, including Google, Yahoo!, and Alltheweb. Due to the space limitation, we only present the experimental results with three data sets from the Ohsumed medical corpus. Similar to the experiments described in the previous subsection, 10 labeled documents are randomly selected to form the training set, and the top 100 documents retrieved by the Web search engines are downloaded to form the set of unlabeled documents.

Figure 8.2 shows the classification accuracy by SVM and the two text categorization methods (Auxi-SVM and Semi-SVM) using different search engines for the three data sets. For almost all the cases, the text categorization methods using the data retrieved from the Internet are able to improve the classification accuracy of the supervised learning method regardless of the difference among Web search engines.The only exception is when we apply Auxi-SVM to the dataset *bacterial vs. virus* using the Web documents retrieved by Alltheweb. This may be attributed to the fact that the Web documents returned by Alltheweb are irrelevant from the topic of musculo and digestive diseases.

Second, among the three Web search engines, Alltheweb leads to the poorest classification accuracy for all the data sets. The most noticeable one is the data set of *bacterial vs. virus*. When applying the auxiliary approach, we find that both Google and Yahoo! are able to improve the classification accuracy of SVM noticeably while the classification accuracy is reduced to around 45% when using Alltheweb. Third, for all the data sets, both text categorization methods achieve similar results when Google or Yahoo! is used as the search engine. Based on the above results, we thus conclude that different search engines can have significant impact on the classification accuracy of semi-supervised text categorization.

8.3 Summary and Future Work

In this chapter, we presented a general framework for semi-supervised text categorization that collects the unlabeled documents via Web search engines and utilizes them to improve the accuracy of supervised text categorization. We proposed a novel discriminative query generation method that is able to identify query words that are both representative and discriminative. We successfully integrated the semi-supervised learning approach with the Web search engines that proves to outperform the other counterpart text categorization methods. Extensive experiments have demonstrated that the proposed semi-supervised text categorization framework can significantly improve the classification accuracy. Specifically, the classification error is reduced by 30% averaged on the nine data sets when using Google as the search engine.

Two important issues deserve our attentions in the future. First, it is interesting to investigate the performance when a meta search engine is used. Second, since search engines would inevitably output some irrelevant documents, it remains as a research question whether further filtering on the returned documents is needed to lift up the performance.

□ **End of chapter.**

Table 8.2: The query terms selected by different query generation methods for the data set *bacterial vs. virus.* The query size is set to be three for all methods.

QG	**Class**	**Query Terms**	**Acc(%)**
SVM	virus bacterial	syndrome manifestation oral pneumonia mycology presentation	29.5
TF	virus bacterial	patient cmv syndrome patient pylori graft	37.6
TF/IDF	virus bacterial	cmv patient role pylori graft percent	70.9
OR	virus bacterial	assess test syndrome associate chemotherapy clear	39.2
DQG	virus bacterial	assess antibody cryptococcus pneumonia abscess organism	69.5

Table 8.3: The query terms selected by different query generation methods for the data set *money vs. trade.* The query size is set to be three for all methods.

QG	**Class**	**Query Terms**	**Acc(%)**
SVM-LW	trade money	trade billion quarter market reserve early	82.7
TF	trade money	said trade billion said bank stg	73.8
TF/IDF	trade money	figure poehl germany stg rate mln	56.4
OR	trade money	year export washington trade billion market	82.9
DQG	trade money	billion trade said bank quarter reserve	82.0

Table 8.4: The retrieval precision of different query generation methods with varied query sizes.

Data set	**musculo vs. digestive**			**grain vs. wheat**		
Query Size	3	4	5	3	4	5
SVM-LW	50.8	62.6	65.0	37.0	36.5	39.6
TF	49.1	58.3	64.0	43.2	48.1	51.8
TF/IDF	49.1	57.2	60.6	39.5	40.3	42.8
OR	43.3	44.2	51.7	37.9	42.8	47.5
DQG	60.9	64.9	65.3	53.2	47.0	45.3

Table 8.5: The classification accuracy (%) semi-supervised text categorization methods

	2 Training Examples per Class			**5 Training Examples per Class**		
Data set	**SVM**	**Auxi-SVM**	**Semi-SVM**	**SVM**	**Auxi-SVM**	**Semi-SVM**
male vs. female	56.5	**65.6**	59.4	47.6	**76.1**	73.1
bacterial vs. virus	72.6	43.7	**72.8**	61.8	77.6	**78.3**
musculo vs. digestive	66.8	**69.7**	67.6	69.9	71.3	**77.0**
fourDisease	10.3	22.3	**51.4**	31.6	38.4	**58.0**
ship vs. trade	75.1	91.5	**95.1**	94.1	95.5	**95.9**
corn vs. wheat	56.6	**65.3**	63.0	69.2	69.0	**71.6**
money vs. trade	49.0	**71.0**	59.5	80.6	88.8	**88.9**
auto vs. motor	62.1	74.4	**77.5**	59.4	69.1	**69.2**
sci	29.2	49.1	**50.2**	35.5	56.1	**56.8**
average	53.1	61.4	**66.3**	61.1	71.3	**74.3**

Chapter 9

Conclusion and Future Work

In this chapter, we provide a summary of the thesis. The thesis consists of two parts: the first part deals with good quality unlabeled data which are used in semi-supervised learning literatures, and the second part deals with general unlabeled data which may not be drawn from the same distribution as the labeled data. In the first part, we first conduct an analysis of the fundamental assumptions of semi-supervised learning following the proposed efficient convex relaxation model of TSVM. We then propose an efficient multiple kernel learning approach and naturally extends it to semi-supervised learning. In the second part, we relax the constraint of the quality of unlabeled data. We first consider a setting that the unlabeled data are only structurally-related and may not share the same labels with the training data. We then consider another setting that totally irrelevant data are mixed with good quality data. Finally, we explore the possibility to actively search for unlabeled data from the Internet for semi-supervised learning with its application to text categorization.

Finally, we present future research perspectives for the proposed models.

9.1 Review of the Journey

Learning from unlabeled data has been an important topic in machine learning. One important technique is semi-supervised learning where there are a huge amount of unlabeled data available drawn from the same distribution as the training data.

Current semi-supervised learning methods are either motivated from the manifold assumption or from the low density assumption. In this thesis, we have conducted a theoretical analysis on the unified view of these two assumptions by using Transductive Support Vector Machine (TSVM) as an example. Our results indicate that both of these assumptions are equivalent in their function: both of them can be regarded as a kind of regularization on the unlabeled data. We further develop an efficient convex relaxation for TSVM by deriving the dual of the SDP relaxation of TSVM. Compared with traditional SDP relaxation of TSVM, the prosed relaxation method provides a tighter approximation and incorporates less parameters in the SDP cone. Empirical studies on benchmark data sets demonstrate that the proposed method is more efficient than the previous semi-definite relaxation method and achieves promising classification results comparing with the state-of-the-art methods. We further extend an efficient multiple kernel learning approach to the semi-supervised setting in order to facilitate the learning machine for automatic identification of the best pair-wised similarity among data points.

As semi-supervised learning requires that there are a large amount of unlabeled data drawn from the same distribution as the training data, it is sometimes difficult to be applied to cases where such unlabeled data are expensive to obtain. However, it is possible to utilize data from other tasks, if these data may contain structural information that is helpful to identify the features of the current task. We then develop a novel Supervised

Self-taught Learning (SSTL) model which manages to find the most appropriate high-level features or representations from the poorly-related unlabeled data under the supervision of the labeled training data. Extensive evaluations on various data sets including character image data and Web text data have shown that our proposed algorithm can improve the classification performance against the traditional Self-taught Learning algorithm and the supervised learning algorithm when the number of the labeled data is limited.

We further consider a more complicated case: the unlabeled data are a mixture of good quality data, and irrelevant data and we do not have the prior knowledge on which data samples are relevant or not. We propose a learning framework that is able to deal with such a case. In contrast to the traditional semi-supervised learning that requires unlabeled data to share the same category labels as the labeled data, our work is significantly different from the recent semi-supervised learning with universum or the Universum Support Vector Machines. As an important contribution, we have successfully formulated this new learning approach as a Semi-definite Programming problem, making it solvable in polynomial time. We have also presented theoretical analysis to justify our model. A series of experiments demonstrate that this novel framework has advantages over semi-supervised learning approaches on both synthetic and real data in many facets.

It is difficult to know how much knowledge we can transfer from the unlabeled data from other relevant tasks. Can we actively find relevant data from the Internet since the Internet can be seen as a huge database for almost any field? Another contribution of this thesis answers the question for text categorization: firstly, we develop a query generation method which can automatically generate queries from a few labeled documents; then, we retrieve the Web using the query words to get a batch of more relevant documents; finally, the retrieved documents can then be used as the unlabeled data for semi-supervised learning techniques. Extensive experiments have demonstrated that the proposed semi-supervised text categorization framework can significantly improve the classification accuracy. Specifically, the classification error is reduced by 30% averaged on several data sets when using Google as the search engine.

9.2 Future Work

This thesis tries to plot a whole picture of research work on learning from general unlabeled data. Following this framework, there is a lot of immediate directions inside the proposed models in this thesis.

Firstly, it is still valuable to mine deeply into the nature of learning from unlabeled data in the semi-supervised learning setting. Although we have shown in the thesis an unified view of the fundamental assumptions as a framework of regularization, we do not answer the question of how much regularization do we need to obtain from the unlabeled data. Or alternatively, can we adapt the regularization with the data? This interesting topic deserves more attentions and remains to be an open problem.

Secondly, although we have proposed an efficient convex relaxation approach for TSVM, it is still difficult to apply in large scale data sets due to its SDP nature. How to efficiently and accurately solve the nonlinear optimization problem incorporated in TSVM is still an open problem which devotes a lot of research effort.

Thirdly, for the work of actively retrieving unlabeled data from the Web, two important issues deserve our attentions in the future. One issue is to investigate the performance when a meta search engine is used. The other is to explore whether further filtering on the returned documents is needed to lift up the performance since search engines would inevitably output some irrelevant documents.

Finally, it is still an open problem regarding how to utilize the weakly-related unlabeled data. The framework of self-taught learning provides a possible solution to empirically use the weakly-related unlabeled data. It is desirable to theoretically understand how much we can obtain from the analysis of the quality of these unlabeled data.

Appendix A

Proof of Theorem 1, Chapter 4

Proof. We prove the convergence rate of the extended level method for MKL following the work of [92].

We denote the diameter of the set $\mathcal{P}$ by $D(\mathcal{P})$ and denote the Lipschitz constant of $f(\mathbf{p}, \alpha)$ with respect to $\mathbf{p}$ by $L_{\mathbf{p}}(f)$.

To show the convergence of Δ^i, we will show the number of steps needed to reduce Δ^i by a factor $1-\lambda$ is bounded. Without loss of generality, we consider the decreasing series $(\Delta^1, \ldots, \Delta^N)$ where N is the largest index such that $\Delta^N \geq \Delta^1(1-\lambda)$ and $\Delta^{N+1} < \Delta^1(1-\lambda)$.

First, we show that $\tilde{\mathbf{p}}^N \in \mathcal{G}^i, i = 1, \ldots, N$. This is equivalent to show that $g^i(\tilde{\mathbf{p}}^N) \leq \ell^i$, which can be proved by applying $g_i(\tilde{\mathbf{p}}^N) \leq \underline{f}^N$ and $\underline{f}^N \leq \overline{f}^N - (1-\lambda)\Delta_1$. Then, based on $\mathbf{p}^{i+1} = \pi_{\mathcal{L}^i}(\mathbf{p}^i)$, i.e., the projection of $\tilde{\mathbf{p}}^i$ to the level set $\mathcal{L}^i$, we have

$$\|\mathbf{p}^{i+1} - \tilde{\mathbf{p}}^N\|_2^2 \leq \|\mathbf{p}^i - \tilde{\mathbf{p}}^N\|_2^2 - \|\mathbf{p}^{i+1} - \mathbf{p}^i\|_2^2 \tag{A.1}$$

The following inequations

$$\begin{aligned} \|\mathbf{p}^1 - \tilde{\mathbf{p}}^N\|_2^2 &\geq \|\mathbf{p}^1 - \mathbf{p}^2\|_2^2 + \|\mathbf{p}^2 - \tilde{\mathbf{p}}^N\|_2^2 && \text{(A.2)} \\ &\geq \sum_{i=1}^{N} \|\mathbf{p}^i - \mathbf{p}^{i+1}\|_2^2 + \|\mathbf{p}^N - \tilde{\mathbf{p}}^N\|_2^2 && \text{(A.3)} \end{aligned}$$

then follow. We have $\sum_{i=1}^{N-1} \|\mathbf{p}^i - \mathbf{p}^{i+1}\|_2^2 \leq D^2(\mathcal{P})$, which indicates that the sum of distance square during the first N steps is bounded by the diameter of region $\mathcal{P}$.

Below, we will show the lower bound for $\sum_{i=1}^{N-1}\|\mathbf{p}^i-\mathbf{p}^{i+1}\|_2^2$ using λ and Δ^1. Based on $g^i(\mathbf{p}^i)=f(\mathbf{p}^i,\alpha^i)$ and $g^i(\mathbf{p}^{i+1})\leq \ell^i$, we have

$$\begin{aligned} g^i(\mathbf{p}^i)-g^i(\mathbf{p}^{i+1}) \geq f(\mathbf{p}^i,\alpha^i)-\ell^i \geq \overline{f}^i-(1-\lambda)\underline{f}^i-\lambda\overline{f}^i \\ \geq (1-\lambda)\Delta^i. \end{aligned}$$

We further have $g^i(\mathbf{p}^i)-g^i(\mathbf{p}^{i+1})\geq (1-\lambda)\Delta^N$. Hence, we have

$$(1-\lambda)^2\Delta^N \leq |g^i(\mathbf{p}^i)-g^i(\mathbf{p}^{i+1})|^2 \leq L_{\mathbf{p}}^2(f)\|\mathbf{p}^i-\mathbf{p}^{i+1}\|_2^2.$$

Therefore, to each $\Delta^{N+1}\leq \Delta^1(1-\lambda)$, we need at most

$$N_1 \leq \frac{D(\mathcal{P})^2L_{\mathbf{p}}^2(f)}{(1-\lambda)^2[\Delta^N]^2} \leq \frac{D(\mathcal{P})^2L_{\mathbf{p}}^2(f)}{(1-\lambda)^4[\Delta^1]^2}.$$

Clearly, in order to reduce Δ^i from $(1-\lambda)^{s-1}\Delta^1$ to $(1-\lambda)^s\Delta^1$ where $1\leq s\leq N$, we need at most

$$N_s \leq \frac{D^2(\mathcal{P})L_{\mathbf{p}}^2(f)}{(1-\lambda)^2(1-\lambda)^{2s}[\Delta^1]^2}.$$

So, the number of steps needed to reduce Δ^i from Δ^1 to $\Delta^1(1-\lambda)^s$ is bounded by $N=\sum_{i=1}^s N_i$ and can be further bounded by

$$\frac{D^2(\mathcal{P})L_{\mathbf{p}}^2(f)}{(1-\lambda)^2(1-\lambda)^{2s}[\Delta^1]^2}\frac{1}{1-(1-\lambda)^2}.$$

To reach the error ε, we have

$$(1-\lambda)^s[\Delta^1]^2 \leq \varepsilon \leq (1-\lambda)^{(s-1)}[\Delta^1]^2.$$

Thus, we have

$$N \leq c(\lambda)D^2(\mathcal{P})L_{\mathbf{p}}^2(f)\frac{1}{\varepsilon^2},$$

where $c(\lambda)=\frac{1}{(1-\lambda)^4\lambda(2-\lambda)}$

We now calculate the $D(\mathcal{P})$ and $L_{\mathbf{p}}(f)$ for the MKL problem. Since $\mathcal{P} = \{\mathbf{p} \in \mathbb{R}^m : \mathbf{p}^\top \mathbf{e} = 1,\ 0 \leq \mathbf{p} \leq 1\}$, we can prove that the diameter of the set $\mathcal{P}$ is equal to

$$D(\mathcal{P}) = \max_{\mathbf{p},\mathbf{p}' \in \mathcal{P}} \|\mathbf{p} - \mathbf{p}'\|_2 = \sqrt{2}.$$

According to the definition of $L_{\mathbf{p}}(f)$, we have

$$\begin{aligned} L_{\mathbf{p}}(f) &= \max_{\mathbf{p}\in\mathcal{P},\alpha\in\mathcal{Q}} \|\nabla f_{\mathbf{p}}(\mathbf{p},\alpha)\|_2 \\ &= \frac{1}{2}\|[(\alpha \circ \mathbf{y})^\top \mathbf{K}_1(\alpha \circ \mathbf{y}), \ldots, (\alpha \circ \mathbf{y})^\top \mathbf{K}_m(\alpha \circ \mathbf{y})]^\top\|_2 \\ &\leq \frac{1}{2} n C^2 \sqrt{m} \max_{1\leq i\leq m} \Lambda_{\max}(\mathbf{K}_i). \end{aligned}$$

The operator $\Lambda_{\max}(M)$ computes the maximum eigenvalue of matrix M.

This ends the proof. □

Appendix B

Publications

Conference papers:

1. Zenglin Xu, Rong Jin, Irwin King, and Michael R. Lyu. An Extended Level Method for Multiple Kernel Learning. In Advances in Neural Information Processing Systems (NIPS 22), accepted, 2008.

2. Zenglin Xu, Rong Jin, Kaizhu Huang, Irwin King, and Michael R. Lyu. Semi-supervised text categorization by active search. Appearing in Proceedings of ACM 17th Conference on Information and Knowledge Management (CIKM 2008), accepted, 2008.

3. Kaizhu Huang, Zenglin Xu, Irwin King, and Michael R. Lyu. Semi-supervised Learning from General Unlabeled Data. Appearing in Proceedings of the 8th IEEE International Conference on Data Mining, 2008.

4. Jianke Zhu, Steven Hoi, Zenglin Xu and Michael R. Lyu. An Effective Approach to 3D Deformable Surface Tracking. Appearing in Proceedings of the 10th European Conference on Computer Vision (ECCV 2008), accepted, 2008.

5. Zenglin Xu, Rong Jin, Jianke Zhu, Irwin King, and Michael R. Lyu. Efficient convex relaxation for transductive sup-

port vector machine. In Advances in Neural Information Processing Systems (NIPS 21), 2007.

6. Zenglin Xu, Kaizhu Huang, Jianke Zhu, Irwin King, and Michael R. Lyu, Kernel Maximum a Posteriori Classification with Error Bound Analysis, in Proceedings of the International Conference on Neural Information Processing (ICONIP2007), Kitakyushu, Japan, 2008, pp. 841-850.
7. Zenglin Xu, Jianke Zhu, Irwin King, and Michael R. Lyu. Maximum margin based semi-supervised spectral kernel learning. In Proceedings of 20th International Joint Conference on Neural Network (IJCNN), 12-17 August 2007.
8. Zenglin Xu, Irwin King, and Michael R. Lyu. Web page classification with heterogeneous data fusion. In Proceedings of the 16th international Conference on World Wide Web (Banff, Alberta, Canada, May 08 - 12, 2007). WWW'07. ACM Press, New York, NY, 1171-1172.

Journal papers and book chapters:

1. Zenglin Xu, Kaizhu Huang, Jianke Zhu, Irwin King, and Michael R. Lyu. A Novel Kernel-based Maximum A Posteriori Classification Method. *Neural Networks*, accepted.
2. Zenglin Xu, Irwin King, and Michael R. Lyu. Feature Selection Based on Minimum Error Minimax Probability Machine, International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, iss. 8, pp. 1279-1292, 2007.
3. Kaizhu Huang, Zenglin Xu, Irwin King, Michael R. Lyu, and Zhangbing Zhou, A Novel Discriminative Naive Bayesian Network for Classification, in Bayesian Network Technologies: Applicatons and Graphical Models, Mittal, A. and Kassim, A., Eds., IGI Publishing, 2007, pp. 1-12.

Bibliography

[1] A. Anagnostopoulos, A. Z. Broder, and K. Punera. Effective and efficient classification on a search-engine model. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 208–217, New York, NY, USA, 2006. ACM.

[2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[3] A. Argyriou, M. Herbster, and M. Pontil. Combining graph laplacians for semi–supervised learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 67–74. MIT Press, Cambridge, MA, 2006.

[4] F. R. Bach. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[5] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML '04: Proceedings of the 21th international conference on Machine learning*, pages 41–48, New York, NY, USA, 2004. ACM.

[6] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval.* Addison Wesley, 1999.

[7] M. Balcan and A. Blum. A pac-style model for learning from labeled and unlabeled data. In *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, pages 111–126, 2005.

[8] M. S. Bazaraa. *Nonlinear Programming: Theory and Algorithms.* New York: Wiley, 2nd edition, 1993.

[9] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of the 17th Annual Conference on Computational Learning Theory (COLT)*, pages 624–638, 2004.

[10] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[11] S. Ben-david and R. Schuller. Exploiting task relatedness for multiple task learning. In *Proceedings of Computational Learning Theory (COLT*, 2003.

[12] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11 (NIPS 11)*, pages 368–374. MIT Press, 1999.

[13] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, Massashusetts, 2nd edition, 1999.

[14] J. Bi, T. Zhang, and K. P. Bennett. Column-generation boosting methods for mixture of kernels. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 521–526, New York, NY, USA, 2004. ACM Press.

[15] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 81–88, New York, NY, USA, 2007. ACM Press.

[16] T. D. Bie and N. Cristianini. Convex methods for transduction. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[17] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 19–26, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[18] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML '01: Proceedings of the 18th international conference on Machine learning*, pages 19–26. Morgan Kaufmann, San Francisco, CA, 2001.

[19] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory (COLT-1998)*, pages 92–100, New York, NY, USA, 1998. ACM Press.

[20] D. Boley, M. Gini, R. Gross, E.-H. S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the World Wide Web using WebACE. *Artificial Intelligence Review*, 13(5-6):365–391, 1999.

[21] J. Bonnans, J. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization, Theoretical and Practical Aspects*. Springer-Verlag, Berlin, 2nd ed., 2006.

[22] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[23] A. Z. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 231–238, New York, NY, USA, 2007. ACM.

[24] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[25] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75.

[26] Y. Chang, M. Kim, and V. V. Raghavan. Construction of query concepts based on feature clustering of documents. *Journal Information Retrieval*, 9(3):231–248, 2006.

[27] O. Chapelle, M. Chi, and A. Zien. A continuation method for semi-supervised SVMs. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 185–192, New York, NY, USA, 2006. ACM Press.

[28] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[29] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[30] O. Chapelle, V. Sindhwani, and S. Keerthi. Branch and bound for semi-supervised support vector machines. In

B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19.* MIT Press, Cambridge, MA, 2007.

[31] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

[32] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[33] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. Advances in Neural Information Processing Systems 15, 2003.

[34] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 57–64, 2005.

[35] S.-L. Chuang and L.-F. Chien. A practical web-based approach to generating topic hierarchy for text segments. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 127–136, New York, NY, USA, 2004. ACM Press.

[36] F. R. Chung. *Spectral graph theory.* the American Mathematical Society, 1997.

[37] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *Journal of Machine Learning Reseaerch*, 7:1687–1712, 2006.

[38] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. In *Neural Information Processing Systems (NIPS 15)*, pages 537–544, 2002.

[39] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola. On kernel-target alignment. In *Neural Information Processing Systems (NIPS 13)*, pages 367–373, 2001.

[40] G. Dai and D.-Y. Yeung. Kernel selection forl semi-supervised kernel machines. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 185–192, New York, NY, USA, 2007. ACM Press.

[41] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Self-taught clustering. In *ICML '08: Proceedings of the 25th international conference on Machine learning.*

[42] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 193–200, New York, NY, USA, 2007. ACM Press.

[43] H. Daumé III. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic, 2007.

[44] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126, 2006.

[45] T. De Bie and N. Cristianini. Semi-supervised learning using semi-definite programming. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 120–135. MIT Press, Cambridge, MA, 2006.

[46] G. Druck, C. Pal, A. McCallum, and X. Zhu. Semi-supervised classification with hybrid genera-

tive/discriminative methods. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–289, New York, NY, USA, 2007. ACM.

[47] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, 2000.

[48] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience Publication, 2000.

[49] M. Dudik, R. Schapire, and S. Phillips. Correcting sample selection bias in maximum entropy density estimation. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 323–330. MIT Press, Cambridge, MA, 2006.

[50] G. W. Flake, E. J. Glover, S. Lawrence, and C. L. Giles. Extracting query modifications from nonlinear svms. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 317–324, New York, NY, USA, 2002. ACM Press.

[51] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, 2nd edition, 1990.

[52] E. Gabrilovich and S. Markovitch. Harnessing the expertise of 70,000 human editors: Knowledge-based feature generation for text categorization. *Journal of Machine Learning Research*, 8:2297–2345, 2007.

[53] R. Ghani, R. Jones, and D. Mladenic. Online learning for web query generation: Finding documents matching a minority concept on the web. In *WI '01: Proceedings of the First Asia-Pacific Conference on Web Intelligence: Research and Development*, pages 508–513, London, UK, 2001. Springer-Verlag.

[54] R. Guzman, M. Montes, P. Rosso, and L. Villasenor. Improving text classification by web corpora. In *Proceedings of the 5th Atlantic Web Intelligence Conference*, pages 154–159, 2007.

[55] S. Hasler, H. Wersing, and E. Korner. Combining reconstruction and discrimination with class-specific sparse coding. *Neural Computation*, 19:1897–1918, 2007.

[56] M. Henzinger, B.-W. Chang, B. Milch, and S. Brin. Query-free news search. *World Wide Web*, 8(2):101–126, 2005.

[57] J.-B. Hiriart-Urruty and C. Lemarechal. *Convex analysis and minimization algorithms II: advanced theory and bundle methods. (2nd part edition)*. Springer-Verlag, New York, 1993.

[58] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web (WWW-2006)*, pages 633–642, New York, NY, USA, 2006. ACM Press.

[59] S. C. H. Hoi, M. R. Lyu, and E. Y. Chang. Learning the unified kernel machines for classification. In *Proceedings of Twentith International Conference on Knowledge Discovery and Data Mining (KDD-2006)*, pages 187–196, New York, NY, USA, 2006. ACM Press.

[60] C. W. Hsu and C. J. Lin. A comparision of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.

[61] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schlkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors,

Advances in Neural Information Processing Systems 19, pages 601–608. MIT Press, Cambridge, MA, 2007.

[62] K. Huang, I. King, and M. R. Lyu. Discriminative training of bayesian chow-liu tree multinet classifiers. In *Proceedings of International Joint Conference on Neural Network (IJCNN-2003), Oregon, Portland, U.S.A.*, volume 1, pages 484–488, 2003.

[63] K. Huang, Z. Xu, I. King, and M. R. Lyu. Semi-supervised text categorization by active search. In *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM2008)*, 2008.

[64] K. Huang, H. Yang, I. King, and M. R. Lyu. Learning large margin classifiers locally and globally. In R. Greiner and D. Schuurmans, editors, *Proceedings of The Twenty-First International Conference on Machine Learning*, pages 401–408, 2004.

[65] K. Huang, H. Yang, I. King, and M. R. Lyu. *Machine Learning: Modeling Data Locally and Gloablly.* Springer Verlag, ISBN 3-5407-9451-4, 2008.

[66] K. Huang, H. Yang, I. King, and M. R. Lyu. Maxi-min margin machine: Learning large margin classifiers globally and locally. *IEEE Transactions on Neural Networks*, 19:260–272, 2008.

[67] K. Huang, H. Yang, I. King, M. R. Lyu, and L. Chan. Minimum error minimax probability machine. *Journal of Machine Learning Research*, 5:1253–1286, 2004.

[68] K. Huang, H. Yang, M. R. Lyu, and I. King. Maxi-min margin machine: Learning large margin classifiers localy and globally. *IEEE Transactions on Neural Networks*, 19(2):260–272, 2008.

[69] S.-Y. Huang, C.-R. Hwang, and M.-H. Lin. Kernel Fisher's discriminant analysis in Gaussian Reproducing Kernel Hilbert Space. Technical report, Academia Sinica, Taiwan, R.O.C., 2005.

[70] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 487–493, Cambridge, MA, USA, 1999. MIT Press.

[71] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[72] T. Jebara. *Machine Learning: Discriminative and Generative*. Kluwer, ISBN 1-4020-7647-9, 2003.

[73] J. Jiang and C. Zhai. A two-stage approach to domain adaptation for statistical classifiers. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 401–410, New York, NY, USA, 2007. ACM.

[74] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998. Springer-Verlag.

[75] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML '99: Proceedings of the 16th international conference on Machine learning*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[76] T. Joachims. Transductive learning via spectral graph partitioning. In *ICML '03: Proceedings of the 20th inter-*

national conference on Machine learning, pages 290–297, 2003.

[77] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA, 2006. ACM Press.

[78] P. Kanani, A. McCallum, and C. Pal. Improving author coreference by resource-bounded information gathering from the web. In *IJCAI '07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 429–434, 2007.

[79] S. Kaski and J. Peltonen. Learning from relevant tasks only. In *Machine Learning: ECML 2007*, pages 608–615, 2007.

[80] S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel Fisher discriminant analysis. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 465–472, New York, NY, USA, 2006. ACM Press.

[81] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces, 2002.

[82] S. Kruk and H. Wolkowicz. General nonlinear programming. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, pages 563–575. Kluwer Academic Publishers, Boston, 2000.

[83] J. Lafferty, G. Lebanon, and T. Jaakkola. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:2005, 2005.

[84] J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 801–808. MIT Press, Cambridge, MA, 2008.

[85] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[86] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[87] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[88] G. R. G. Lanckriet, L. E. Ghaoui, and M. I. Jordan. Robust novelty detection with single-class mpm. In *Advances in Neural Information Processing Systerm (NIPS 15)*, 2002.

[89] J. Leblet and M. Quafafou. A new method for query generation applied to learning text classifiers. In *WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, page 633, Washington, DC, USA, 2003. IEEE Computer Society.

[90] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

[91] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 82–90, 2007.

[92] C. Lemaréchal, A. Nemirovski, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69(1):111–147, 1995.

[93] C. Li, J. R. Wen, and H. Li. Text classification using stochastic keyword generation. In *ICML '03: Proceedings of the 20th international conference on Machine learning*, pages 464–471, 2003.

[94] M. Li and Z.-H. Zhou. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[95] X. Liao, Y. Xue, and L. Carin. Logistic regression with an auxiliary data source. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 505–512, New York, NY, USA, 2005. ACM Press.

[96] Z. Liu and W. W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Journal Information Retrieval*, 10(2):173–202, 2007.

[97] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[98] O. L. Mangasarian. *Nonlinear programming.* Philadelphia : Society for Industrial and Applied Mathematics, 1994.

[99] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Appl. Math.*, 123(1-3):397–446, 2002.

[100] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

[101] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Network for Signal Processing Workshop*, pages 41–48, 1999.

[102] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[103] H. Narayanan, M. Belkin, and P. Niyogi. On the relation between low density separation, spectral clustering and graph cuts. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1025–1032. MIT Press, Cambridge, MA, 2007.

[104] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons Ltd, 1983.

[105] Y. Nesterov and A. Nemirovsky. *Interior point polynomial methods in convex programming: Theory and applications*. Studies in Applied Mathematics. Philadelphia, 1994.

[106] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2-3):103–134, 2000.

[107] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, (381):607–609, 1996.

[108] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

[109] C. S. Ong, A. J. Smola, and R. C. Williamson. Learning the kernel with hyperkernels. *Journal of Machine Learning Research*, 6:1043–1071, 2005.

[110] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 683–690, New York, NY, USA, 2007. ACM.

[111] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14*, 1998.

[112] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 759–766, New York, NY, USA, 2007. ACM Press.

[113] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of International Conference on Machine Learning (ICML-2007)*, 2007.

[114] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 713–720, New York, NY, USA, 2006. ACM Press.

[115] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[116] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. To transfer or not to transfer. In *Proceedings*

of NIPS 2005 Workshop on Inductive Transfer: 10 Years Later, 2005.

[117] S. T. Roweis and L. K. Saul. Nonlinear dimensional-ity reduction by locally linear embedding. *Science*, (290):123–137, 2000.

[118] J. Schmidhuber. On learning how to learn learning strategies. Technical report, Technical Report FKI-198-94, Fakultat fur Informatik., 1994.

[119] B. Scholkopf and A. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[120] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[121] J. Shawe-Taylor and Y. Singer, editors. *Regularization and Semi-supervised Learning on Large Graphs*, volume 3120 of *Lecture Notes in Computer Science*. Springer, 2004.

[122] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. A comparison of implicit and explicit links for web page classification. In *Proceedings of the 15th international conference on World Wide Web (WWW 2006)*, pages 643–650, New York, NY, USA, 2006. ACM Press.

[123] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.

[124] V. Sindhwani, S. S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 841–848, New York, NY, USA, 2006. ACM Press.

[125] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831, New York, NY, USA, 2005. ACM Press.

[126] F. H. Sinz, O. Chapelle, A. Agarwal, and B. Scholkopf. An analysis of inference with the universum. In *Advances in Neural Information Processing Systems (NIPS-07)*.

[127] A. Smola and R. Kondor. Kernels and regularization on graphs, 2003.

[128] A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, Neuro-COLT2, 1998.

[129] A. Smola, S. V. N. Vishwanathan, and Q. Le. Bundle methods for machine learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1377–1384. MIT Press, Cambridge, MA, 2008.

[130] A. J. Smola, P. L. Bartlett, B. Scholkopf, and D. Schuurmans. *Advances in Large Margin Classifiers.* The MIT Press, 2000.

[131] S. Sonnenburg, G. Rätsch, and C. Schäfer. Learning interpretable SVMs for biological sequence classification. In *RECOMB*, pages 389–407, 2005.

[132] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[133] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[134] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems (NIPS)*, 1996.

[135] S. Thrun and T. Mitchell. Learning one more thing. In *IJCAI '95: Proceedings of the 14th international joint conference on Artificial Intelligence*, pages 1217–1223, 1995.

[136] I. W.-H. Tsang and J. T.-Y. Kwok. Efficient hyperkernel learning using second order conic programming. *IEEE Transactions On Neural Networks*, 17(1):48–58, 2006.

[137] H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[138] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[139] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition, 1999.

[140] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik. Inference with the universum. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1009–1016, New York, NY, USA, 2006. ACM.

[141] C. K. I. Williams and D. Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

[142] C. K. I. Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. In *Pro-*

ceedings of 17th International Conf. on Machine Learning (ICML-2000), pages 1159 – 1166. Morgan Kaufmann, San Francisco, CA, 2000.

[143] P. Wu and T. G. Dietterich. Improving svm accuracy by training on auxiliary data sources. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 110, New York, NY, USA, 2004. ACM Press.

[144] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, 2004.

[145] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *AAAI*, pages 904–910, 2005.

[146] R. Xu and D. I. Wunsch. Survey of clustering algorithms. *IEEE Transactions On Neural Networks*, 16(3):645–678, 2005.

[147] Z. Xu, R. Jin, K. Huang, I. King, and M. R. Lyu. Semi-supervised text categorization by active search. In *CIKM '08: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 1517–1518, New York, NY, USA, 2008. ACM Press.

[148] Z. Xu, R. Jin, J. Zhu, I. King, and M. R. Lyu. Efficient convex relaxation for transductive support vector machine. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1641–1648. MIT Press, Cambridge, MA, 2008.

[149] Z. Xu, I. King, and M. R. Lyu. Feature selection based on minimum error minimax probability machine. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(8):1–14, 2007.

[150] Z. Xu, I. King, and M. R. Lyu. Web page classification with heterogeneous data fusion. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1171–1172, New York, NY, USA, 2007. ACM Press.

[151] Z. Xu, J. Zhu, I. King, and M. Lyu. Kernel maximum a posteriori classification with error bound analysis. In *Proceedings of the International Conference on Neural Information Processing (ICONIP2007)*, pages 841–850, 2008.

[152] Z. Xu, J. Zhu, M. R. Lyu, and I. King. Maximum margin based semi-supervised spectral kernel learning. In *IJCNN'07: Proceedings of 20th International Joint Conference on Neural Network*, pages 418–423, 2007.

[153] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, page 189C196, 1995.

[154] J. Ye, J. Chen, and S. Ji. Discriminant kernel and regularization parameter learning via semidefinite programming. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1095–1102, New York, NY, USA, 2007. ACM.

[155] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 114, New York, NY, USA, 2004. ACM.

[156] S. Zelikovitz and M. Kogan. Using web searches on important words to create background sets for LSI classification. In *19th FLAIRS conference*, 2006.

[157] D. Zhang, J. Wang, F. Wang, and C. Zhang. Semi-supervised classification with universum. In *SDM*, pages 323–333, 2008.

[158] J. Zhang, R. Jin, Y. Yang, and A. G. Hauptmann. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In *ICML '03: Proceedings of the 20th international conference on Machine learning*, pages 888–895, 2003.

[159] T. Zhang and R. Ando. Analysis of spectral kernel design based semi-supervised learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems (NIPS 18)*, pages 1601–1608. MIT Press, Cambridge, MA, 2006.

[160] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[161] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14:185–205(21), 2005.

[162] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *Advances in Neural Information Processing Systems (NIPS 16)*, 2003.

[163] X. Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.

[164] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic

functions. In *ICML '03: Proceedings of the 20th international conference on Machine learning*, pages 912–919, 2003.

[165] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS 17)*, pages 1641–1648, Cambridge, MA, 2005. MIT Press.

[166] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, New York, NY, USA, 2007. ACM.